Pattern Recognition

(EE5907)

Song Bai

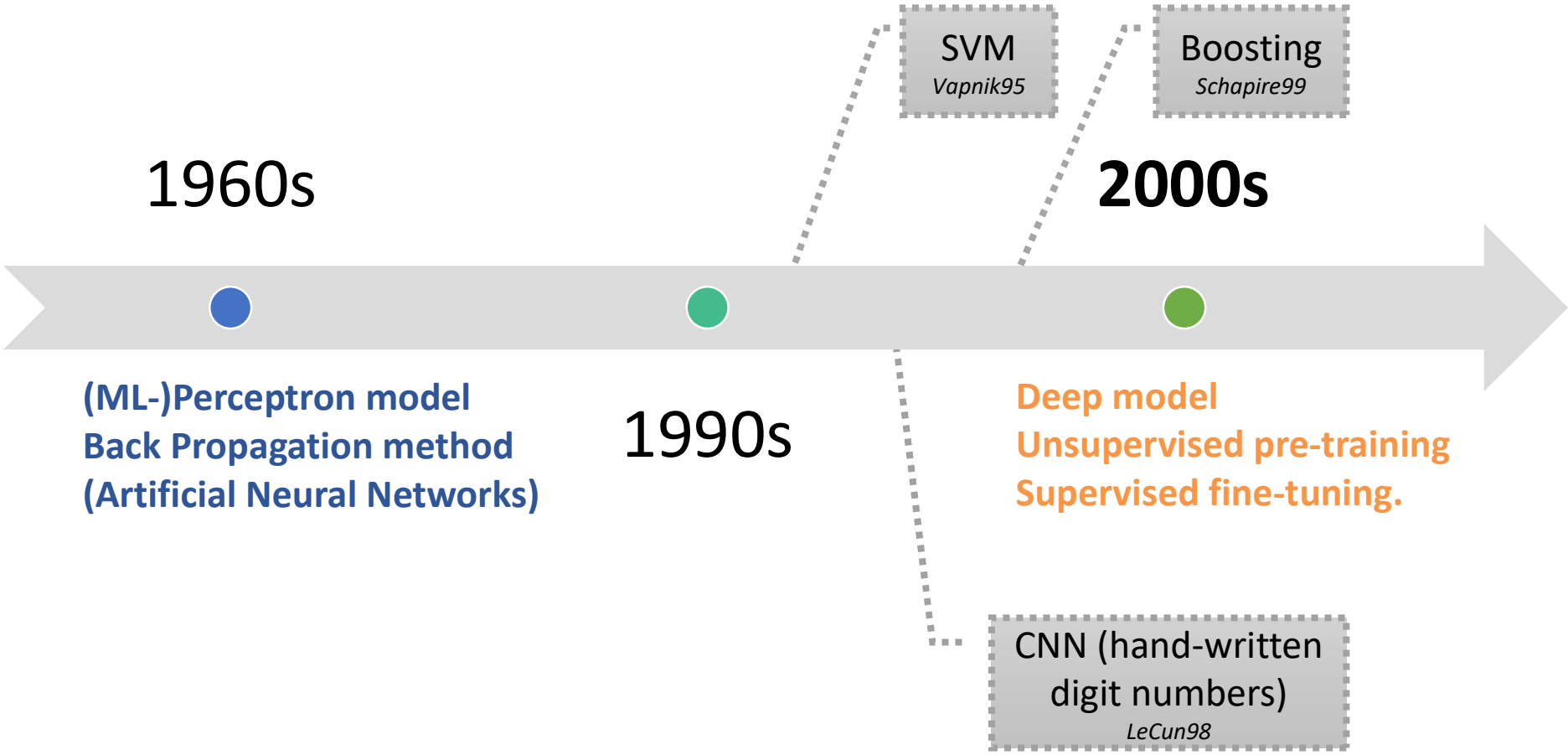
Email: songbai.site@gmail.com

Outlines

- Unsupervised Feature Extraction (PCA, NMF,…)
- Supervised Feature Extraction (LDA, GE, …)
- Clustering and Applications
- Gaussian Mixture Model and Boosting
- Support Vector Machine
- **Deep Learning**

Introduction

- Deep Learning History



Acknowledgement

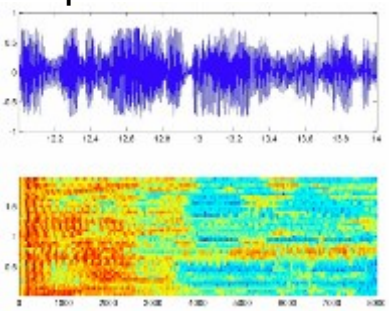
- This presentation is heavily based on:
  - http://cs.nyu.edu/~fergus/pmwiki/pmwiki.php
  - http://deeplearning.net/reading-list/tutorials/
  - http://deeplearning.net/tutorial/lenet.html
  - http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial
- … and many other

Data Explosion

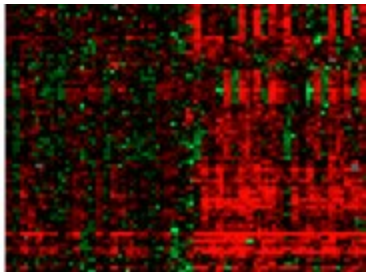
Image & Video

Text & Language

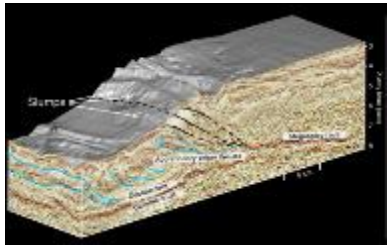
Speech & Audio



Gene Expression

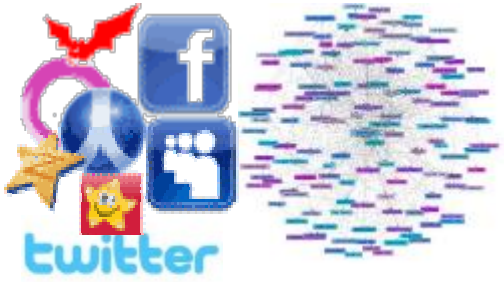


Geological Data

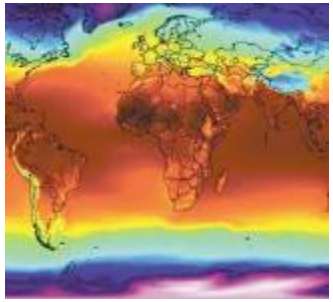


Product Recommendation

Relational Data/ Social Network

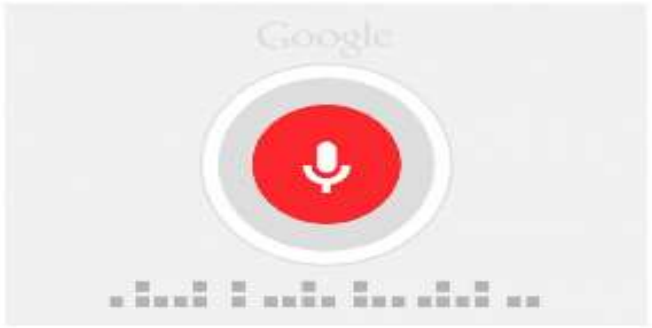


Climate Change



Why Deep Learning?

- End-to-end learning for many tasks.

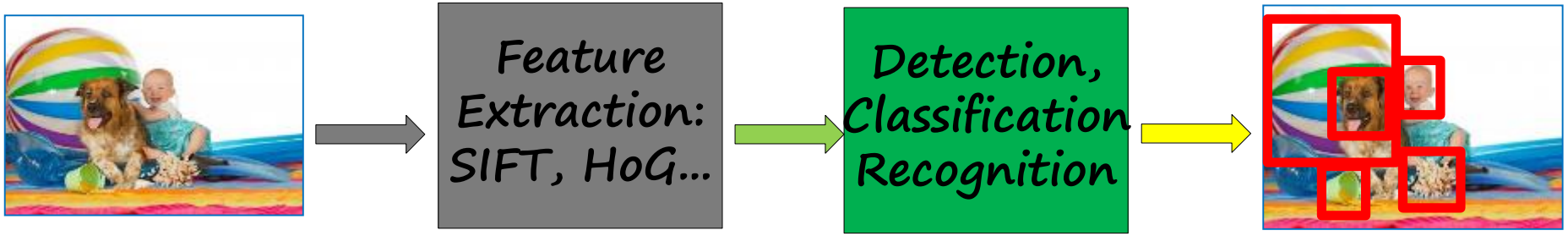


Classical Computer Vision Pipeline

- Given an image

Classical Computer Vision Pipeline.

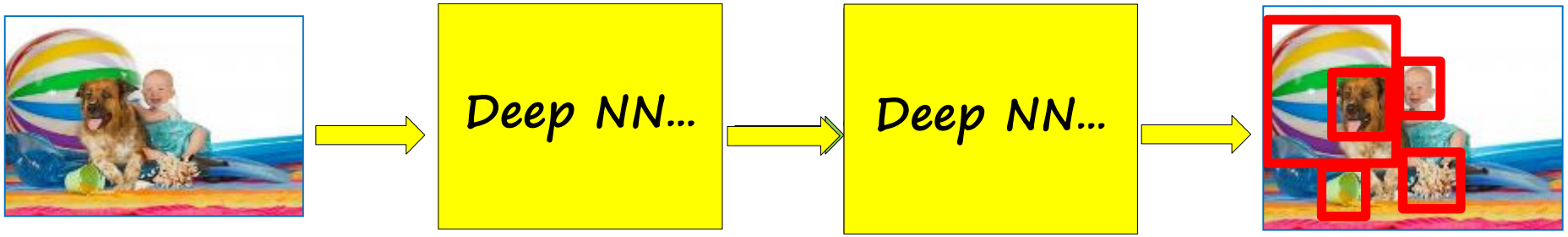
- CV experts
  - Select / develop features: SURF, HoG, SIFT, RIFT, …
  - Add on top of this Machine Learning for multi-class recognition and train classifier



Classical CV feature definition is domain-specific and time-consuming

Deep Learning –based Vision Pipeline.

- Deep Learning:
  - Build features automatically based on training data
  - Combine feature extraction and classification
- DL experts: define NN topology and train NN



Deep Learning promise:
train good feature automatically,
same method for different domain

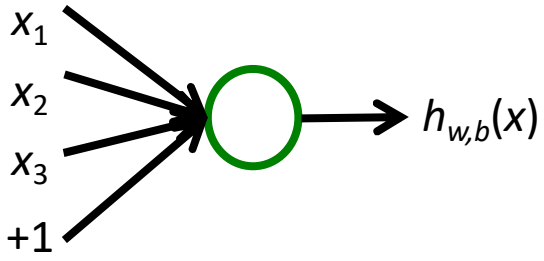
Current Trend in AI

	Artificial Intelligence
Dimension	**High-dimensional** data (e.g. more than 100 dimensions)
Noises	The noise is not sufficient to obscure the **structure** in the data if we process it right.
Structure	There is a **huge amount of structure** in the data, but the structure is too complicated to be represented by a simple model.
Main Task	The main problem is figuring out a way to **represent the complicated structure** so **that it can be learned**.

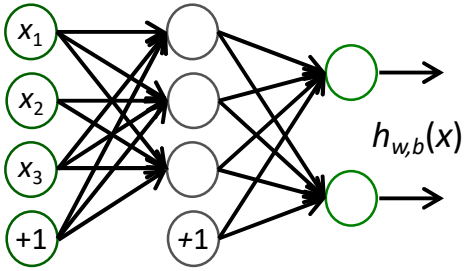
Recent Progresses on DL

- Academic Perspective
  - **Layered pre-training**, Hinton et al. "Reducing the dimensionality of data with neural networks" Science 2006
  - **Speech recognition**: Deep neural networks for acoustic modelling in speech recognition, Hinton, et al, ISPM, 2012
  - **Natural Language Process (NLP)**: Mikolov Tomáš: Statistical Language Models based on Neural Networks. PhD thesis, 2012
  - **Computer Vision**: Deep Learning wins the ImageNet 2012 (**74%** to **85%**), to ImageNet 2014, **93.3%** from Google.
  - Deep Learning appears in all major conferences in machine learning, computer vision, multimedia, speech, text, data mining.

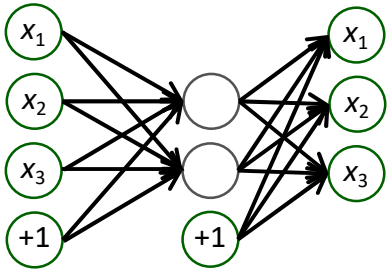
Deep Learning Models



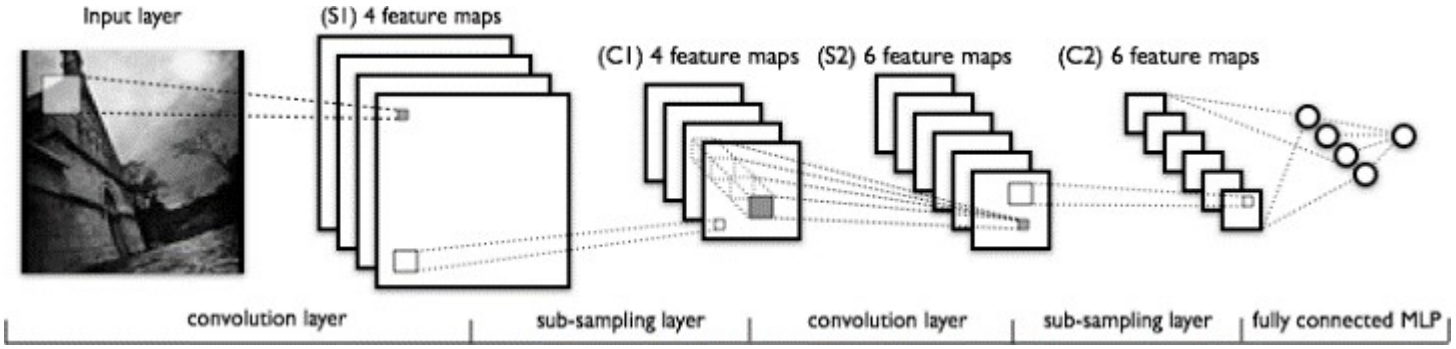
Perceptron Model



Neural Network



Auto-Encoder

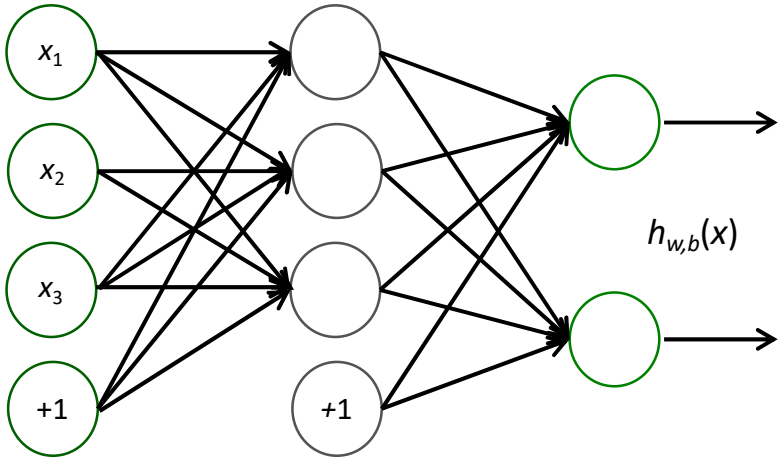


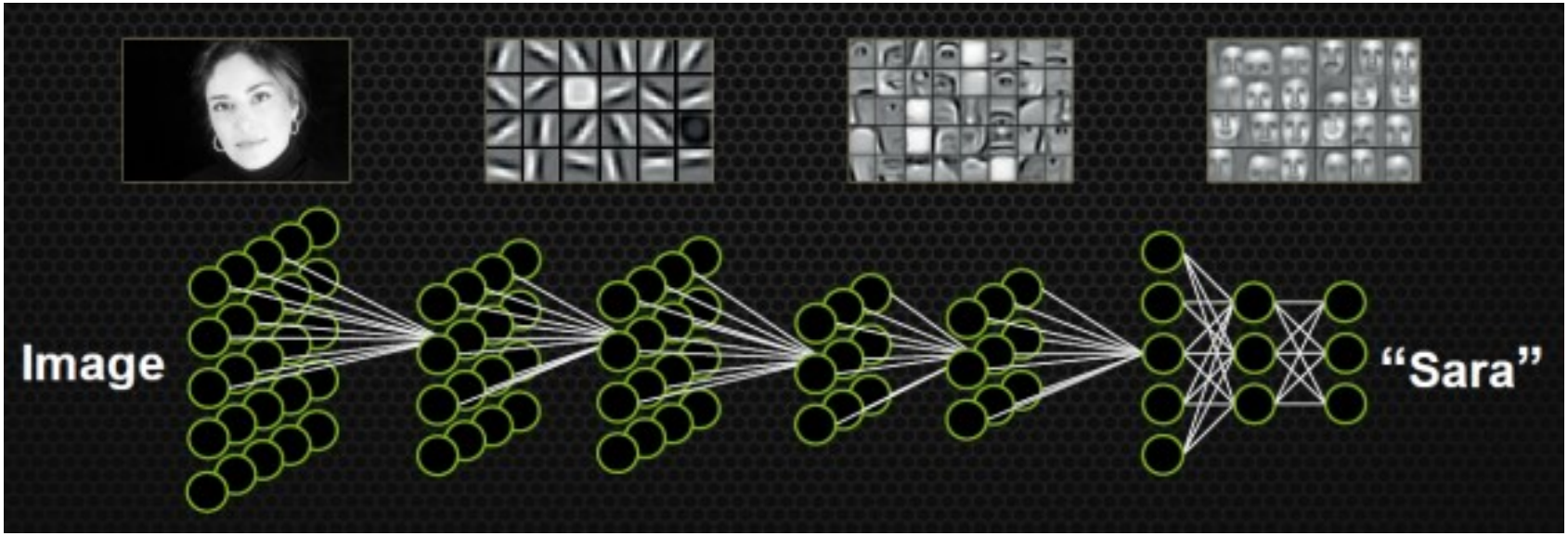
Convolutional Neural Networks

AND MANY MORE …

Neural Networks

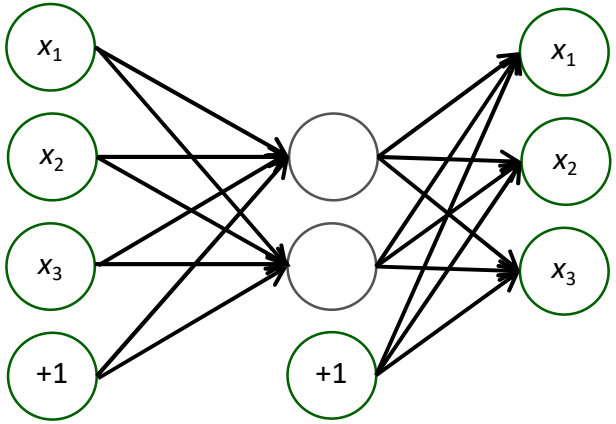
- Structure
  - Input Layer
  - Hidden Layer
  - Output Layer
- Calculation
  - Forward Propagation
    - Testing one input sample
  - Backward Propagation
    - Learning the parameters





Auto-encoder

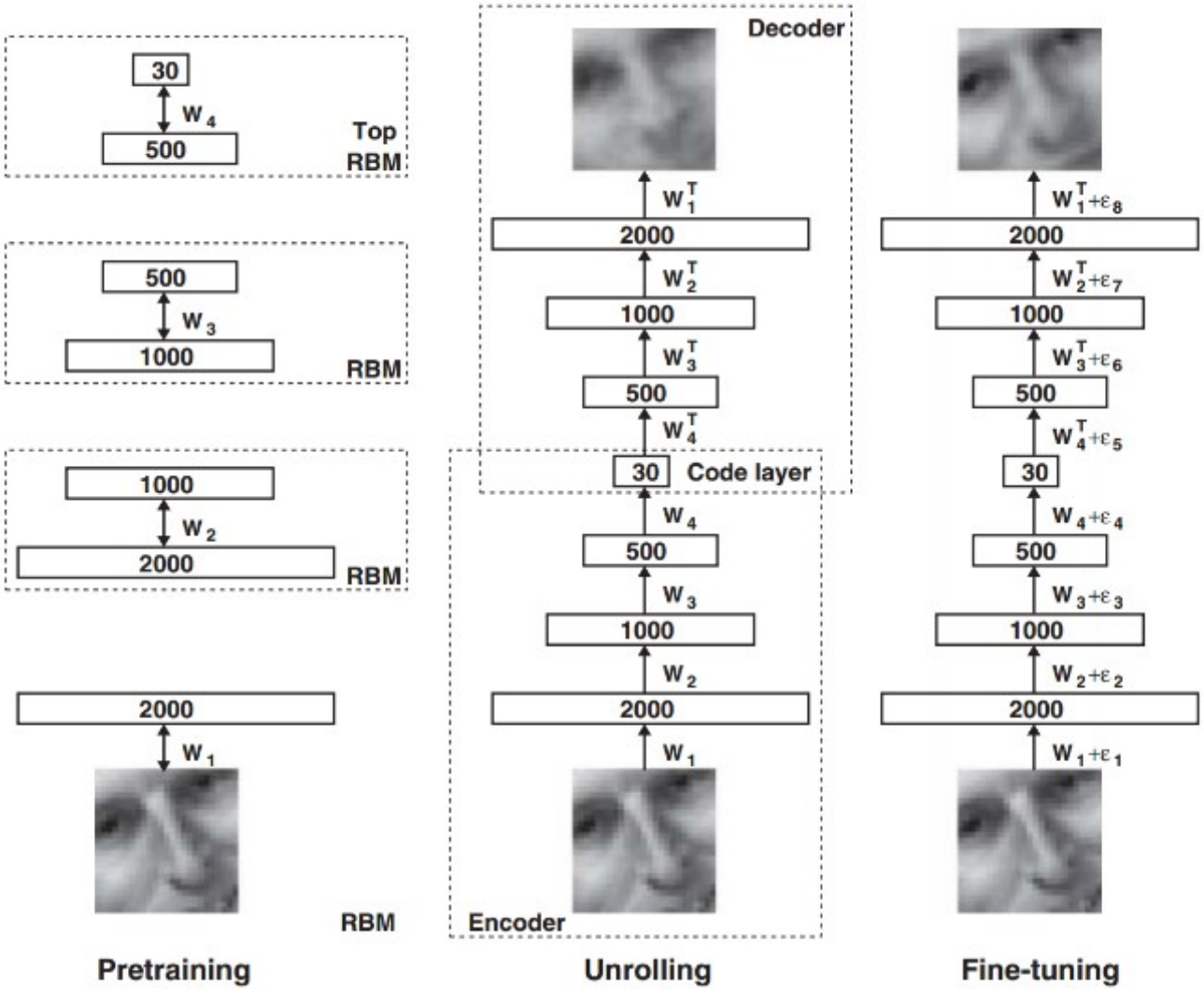
- A neural network with a special structure
  - Input and output layer have the **same** form
  - A number of hidden layers with **smaller** sizes
  - Used for coding or dimensionality reduction
  - Many different variants

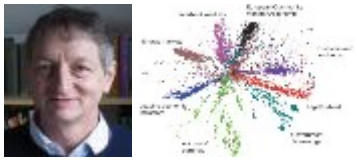


$$\mathbf{x} \approx h_{W,b}(\mathbf{x}) = \sigma(W\mathbf{x} + b)$$

Application to Dimension Reduction

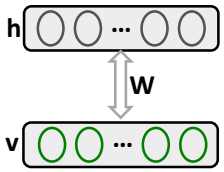
- Hinton's work (2006)





Hinton Science 2006

Dimension reduction like PCA

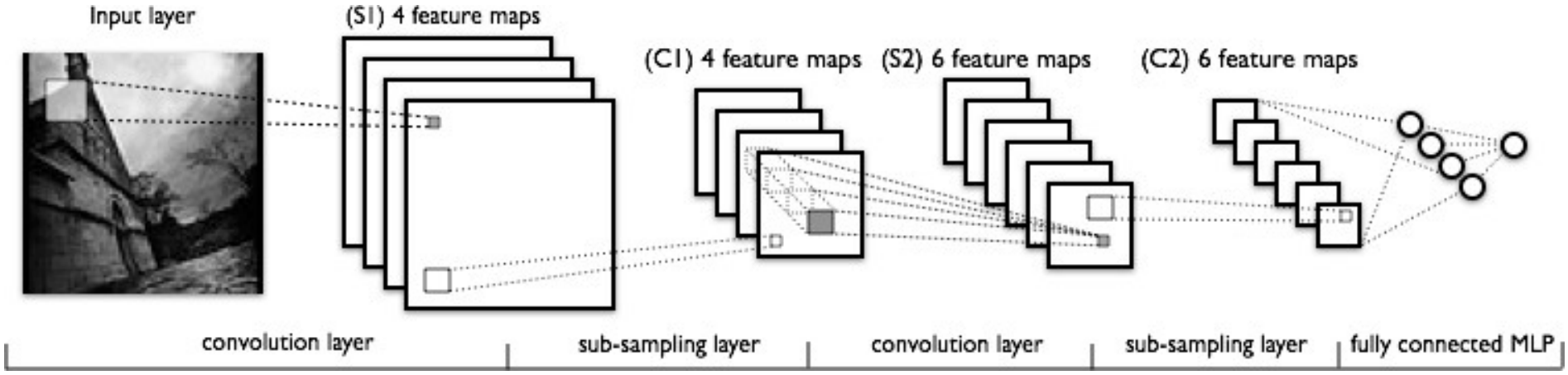


Restricted Boltzmann Machines

Convolutional Neural Networks

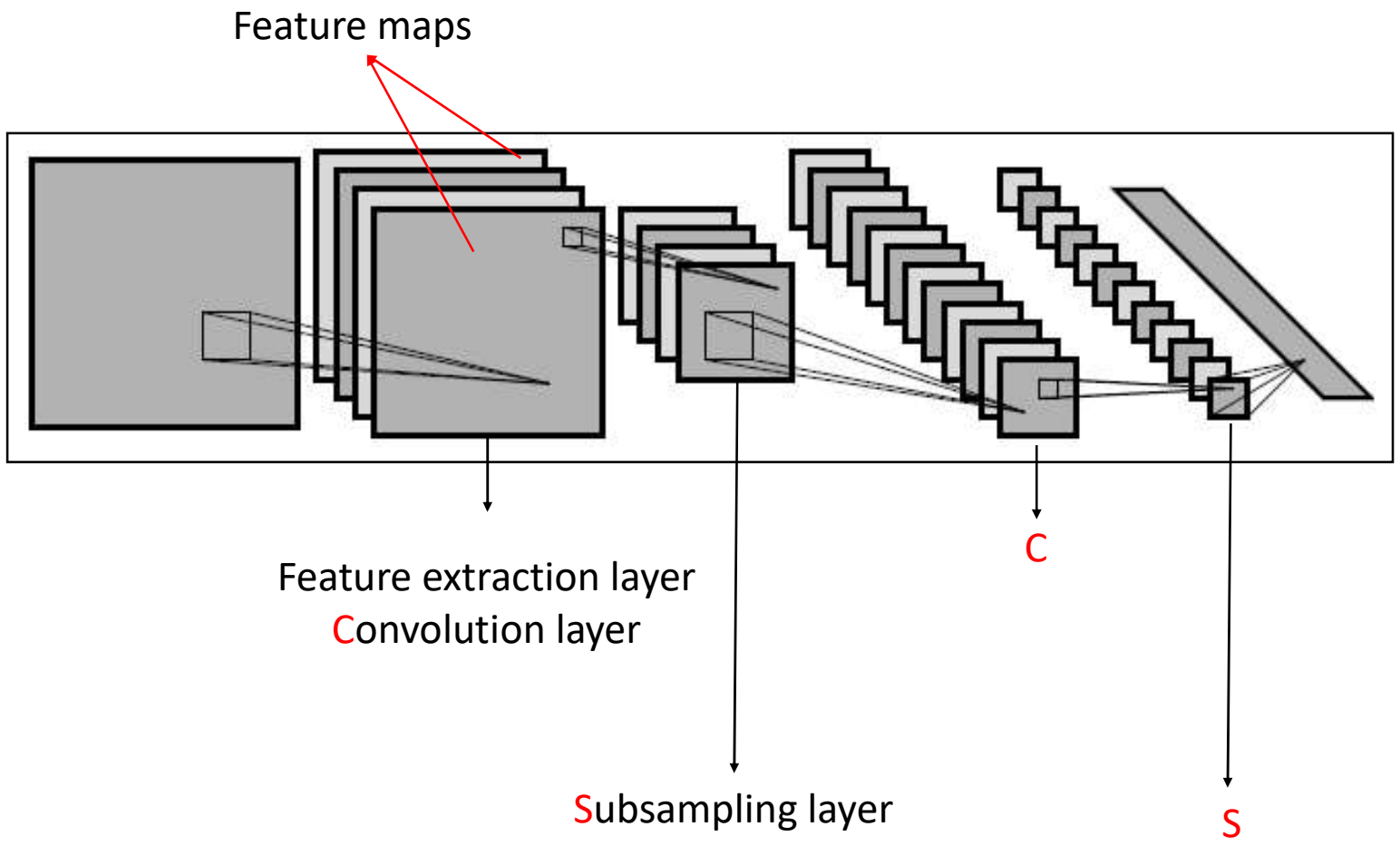
- 卷积神经网络 (Lecun98)
- 以前的神经网络的弊端
- 对移位、缩放和其他形式的失真很敏感
- 忽略了输入的拓扑结构或结构信息
- CNN的特点
- 一个可以从图像中提取拓扑学特性的前馈网络
- 旨在直接从像素图像中识别视觉模式，只需进行最少的预处理

- Convolutional Neural Networks (Lecun98)
  - Drawbacks of previous neural networks
    - Sensitive to shifting, scaling, and other forms of distortion
    - Ignore the topology or structure information of the input
  - Characteristics of CNN
    - A **feed-forward** network that can extract topological properties from an image
    - Designed to **recognize visual patterns** directly from pixel images with minimal preprocessing



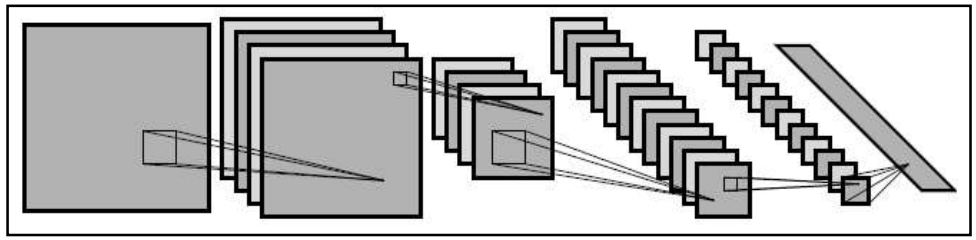
Convolutional Neural Networks

- CNN processing pipeline

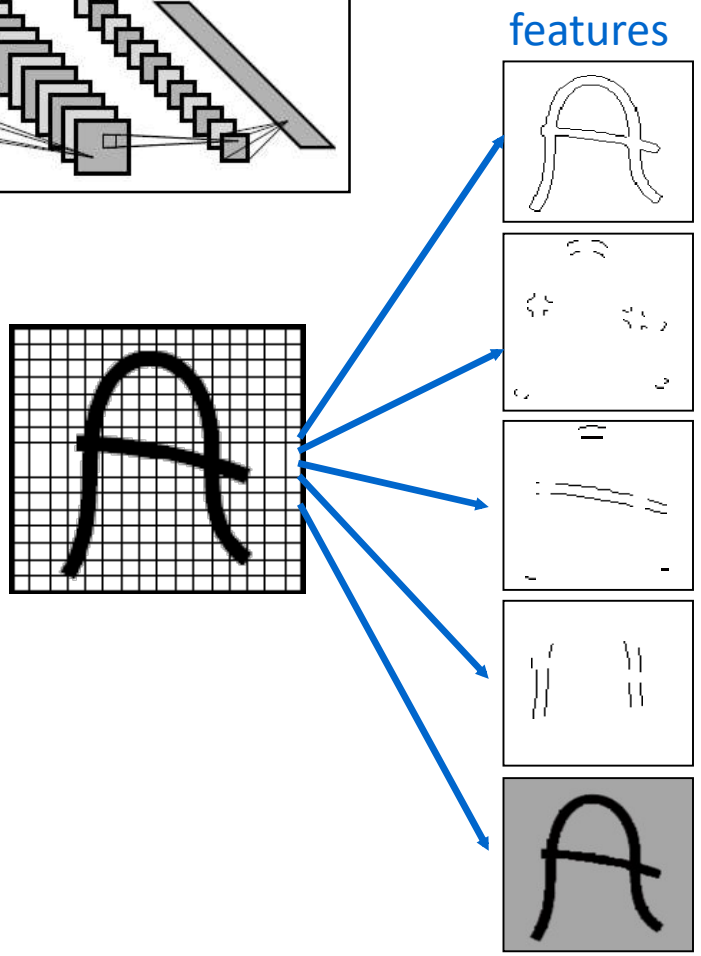


Convolutional Neural Networks

- Feature extraction layer or Convolution layer
  - Detect the same feature at different positions in the input image.

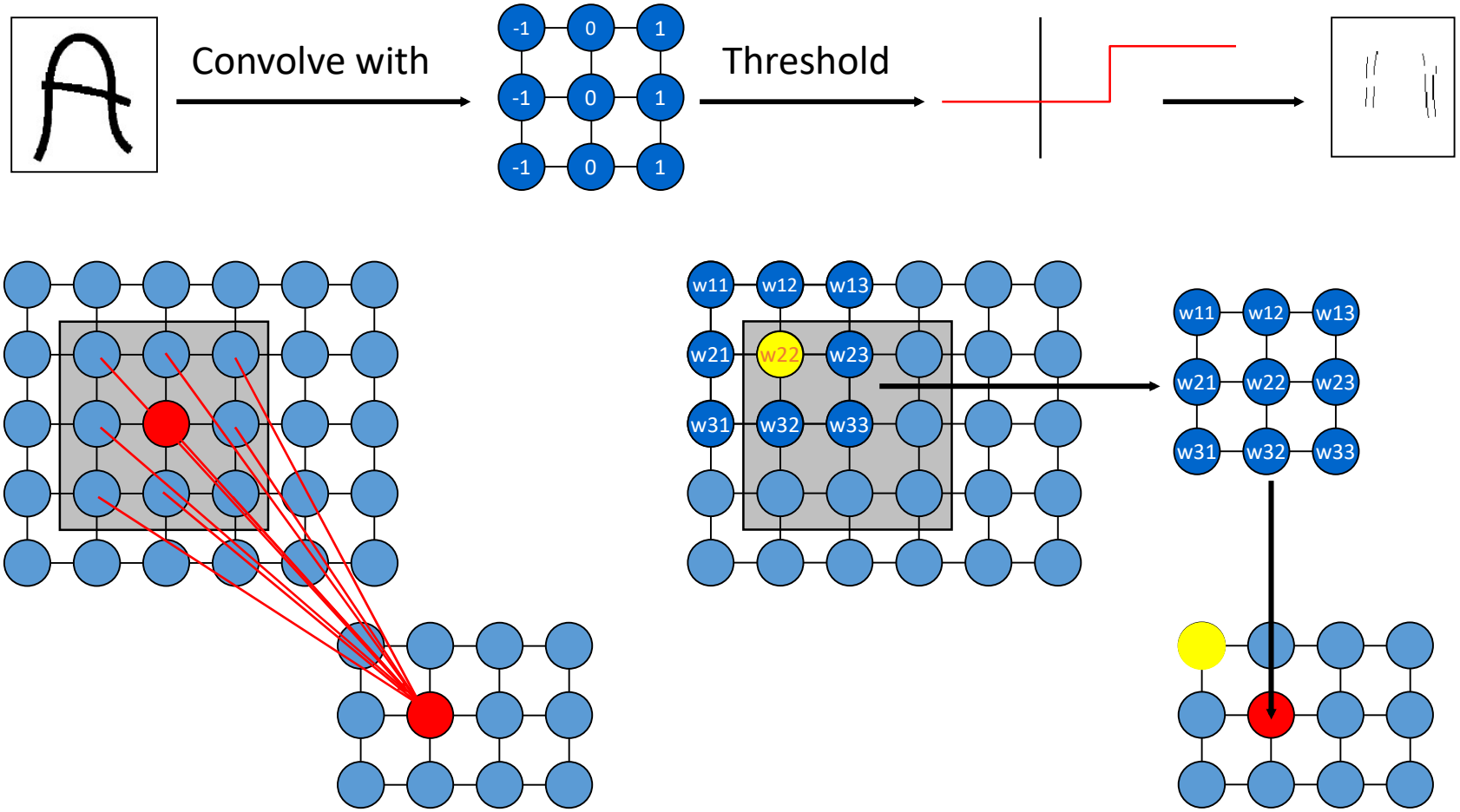


- 特征提取层或卷积层
- 检测输入图像中不同位置的相同特征。



Convolutional Neural Networks

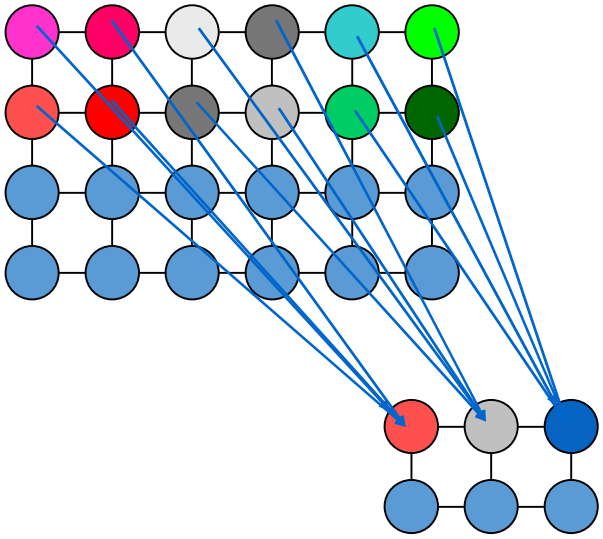
- Feature extraction illustration

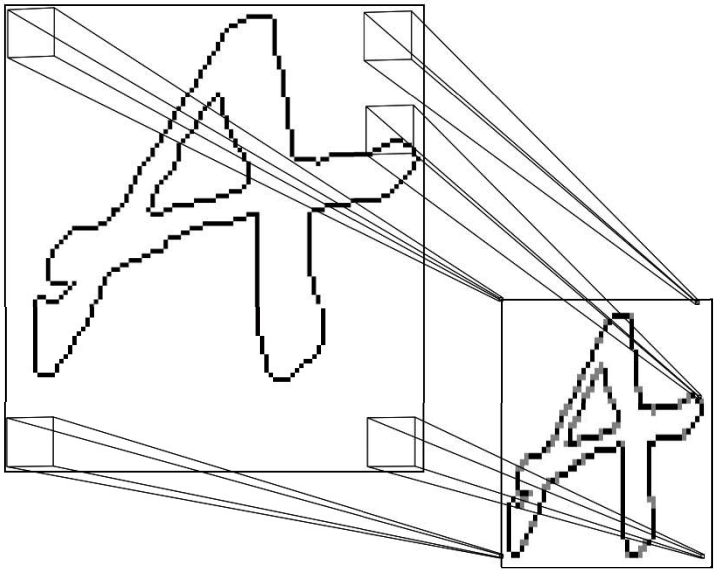


- 子采样层
- 子采样层降低了每个特征图的空间分辨率
- 通过降低特征图的空间分辨率，可以实现一定程度的位移和失真不变性
- 同时，特征的维度也被降低
- 对于不同的任务，子采样策略可能是不同的

Convolutional Neural Networks

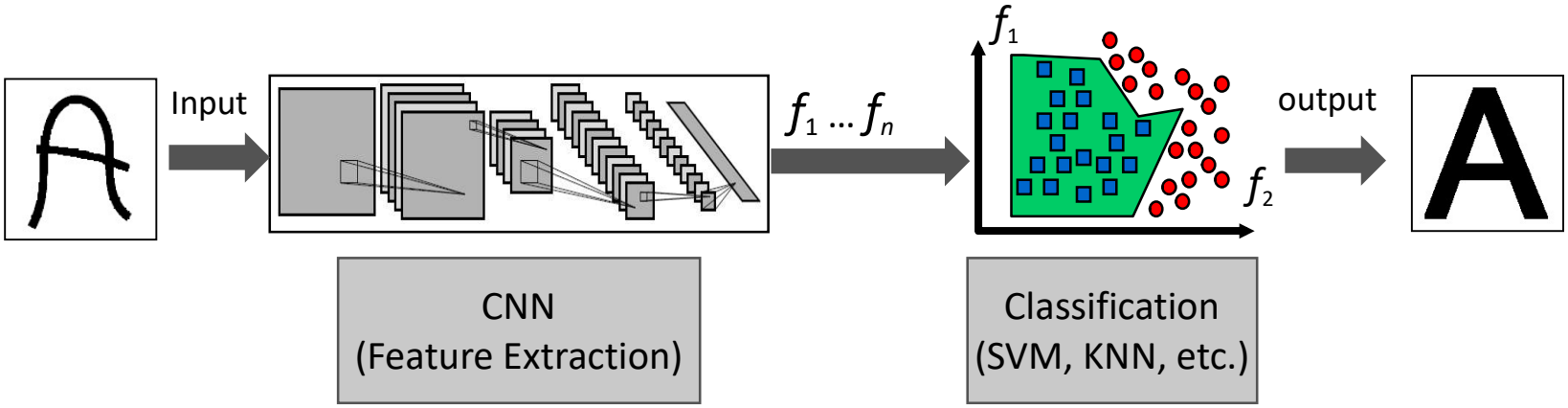
- Subsample layer
  - The **subsampling** layers reduce the spatial resolution of each feature map
  - By reducing the **spatial resolution** of the feature map, a certain degree of shift and distortion invariance is achieved
  - Also the dimension of the feature is reduced
  - Subsampling strategy may be different for different task





Convolutional Neural Networks

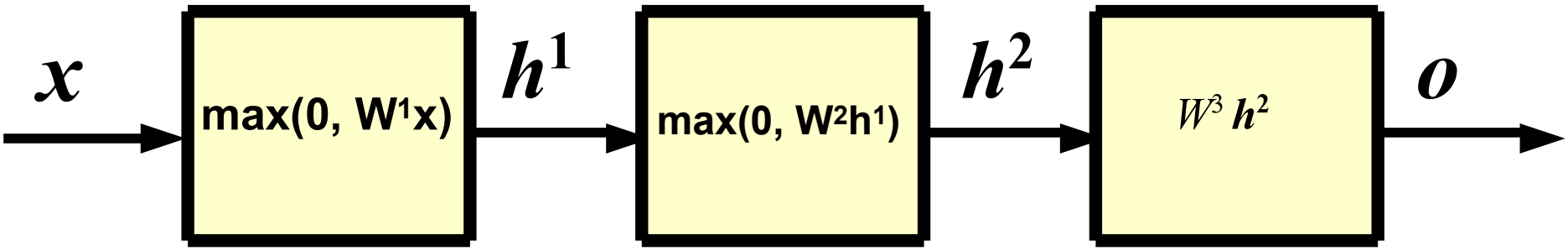
- Classification based on CNN extracted features



Overall Learning Algorithms for DL

- Unsupervised Pre-training
  - Use unlabeled samples to train a distribution or encoder to represent the samples
  - In a layer-wise greedy manner
  - Provide a good initialization of the network
- Supervised Fine-tuning
  - Use labeled samples to further improve accuracy
  - Adapt the network to the target label space and the task

- 无监督的预训练
- 使用未标记的样本来训练一个分布或编码器来代表样本
- 以逐层贪婪的方式
- 为网络提供一个良好的初始化
- 监督下的微调
- 使用有标签的样本来进一步提高准确性
- 使网络适应目标标签空间和任务



x input

h^1 1-st layer hidden units

h^2 2-nd layer hidden units

o output

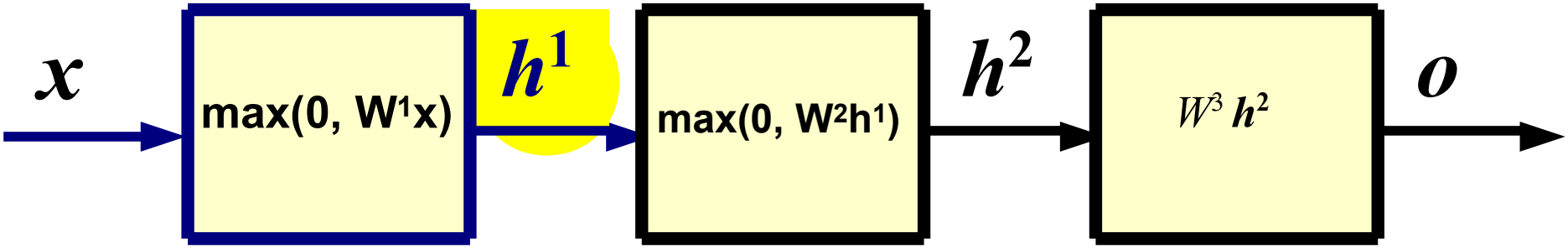
Example of a 2 hidden layer neural network (or 4 layer network, counting also input and output).

Credit: Slides are from M. Ranzato

Forward Propagation

Def.: Forward propagation is the process of computing the output of the network given its input.

定义：前向传播是计算网络输入的输出的过程。



$$x \in R^D \qquad W^1 \in R^{N_1 \times D} \qquad b^1 \in R^{N_1} \qquad h^1 \in R^{N_1}$$

$$h^1 = max(0, W^1 x + b^1)$$

W^1 1-st layer weight matrix or

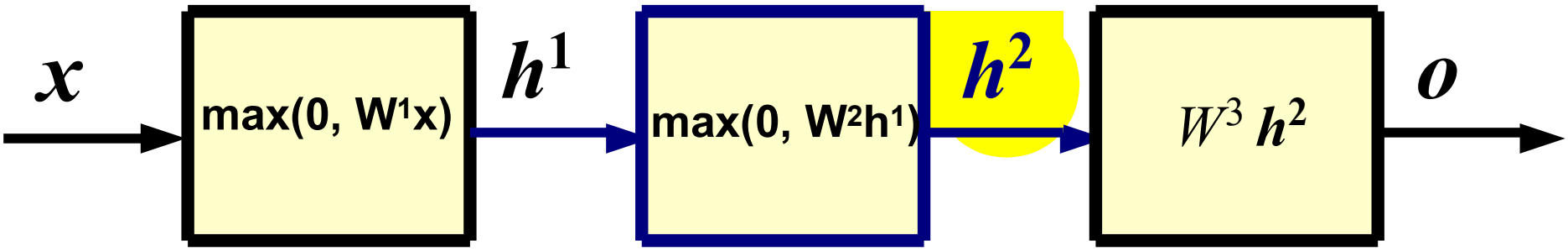
b^1 weights 1-st layer biases

非线性u = max(0,v)在DL文献中被称为ReLU。
每个输出隐藏单元将前一层的所有单元作为输入：每个这样的层被称为 "全连接"。

The non-linearity u = max(0,v) is called ReLU in the DL literature.

Each output hidden unit takes as input all the units at the previous layer: each such layer is called "fully connected".

Credit: Slides are from M. Ranzato



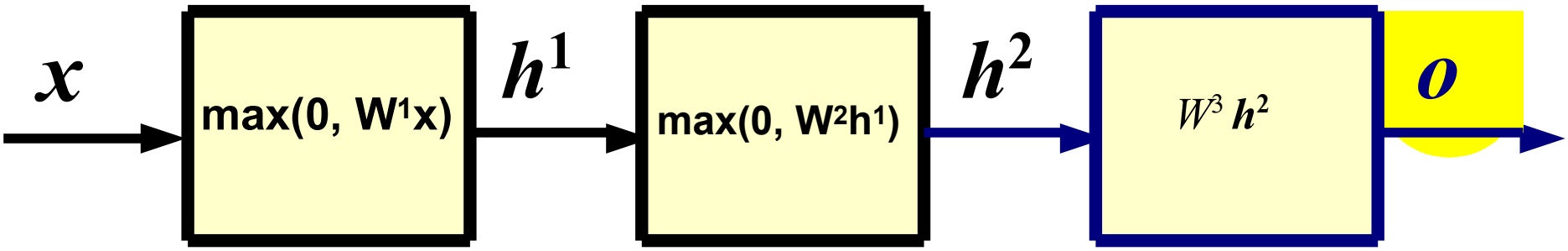
$$h^1 \in R^{N_1} \quad W^2 \in R^{N_2 \times N_1} \quad b^2 \in R^{N_2} \qquad h^2 \in R^{N_2}$$

$$h^2 = max(0,\ W^2 x + b^2)$$

W^2 **2-nd layer weight matrix or weights**

b^2 **2-nd layer biases**

Credit: Slides are from M. Ranzato



$$h^2 \in R^{N_2} \quad W^3 \in R^{N_3 \times N_2} \quad b^3 \in R^{N_3} \qquad o \in R^{N_3}$$

$$o = max(0, W^3 h^2 + b^3)$$

W^3 3-rd layer weight matrix or weights

b^3 3-rd layer biases

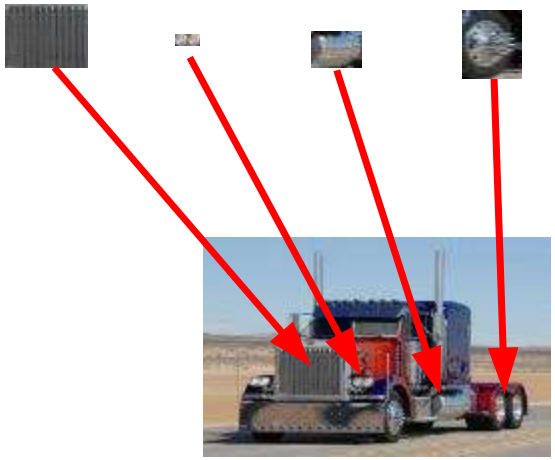
Credit: Slides are from M. Ranzato

问题。为什么我们需要很多层？
解答：当输入有层次结构时，使用分层结构可能更有效率，因为中间的计算可以重复使用。当输入有分层结构时，使用分层架构有可能更有效率，因为中间的计算可以被重复使用。DL架构之所以高效，还因为它们使用分布式表示法，这些表示法在各个类之间共享。

Question: Why do we need many layers?

Answer: When input has hierarchical structure, the use of a hierarchical architecture is potentially more efficient because intermediate computations can be re-used. DL architectures are efficient also because they use distributed representations which are shared across classes.

[0 0 1 0 0 0 0 1 0 0 1 1 0 0 1 0 …] **truck feature**



Exponentially more efficient than a 1-of-N representation (e.g. k-means)

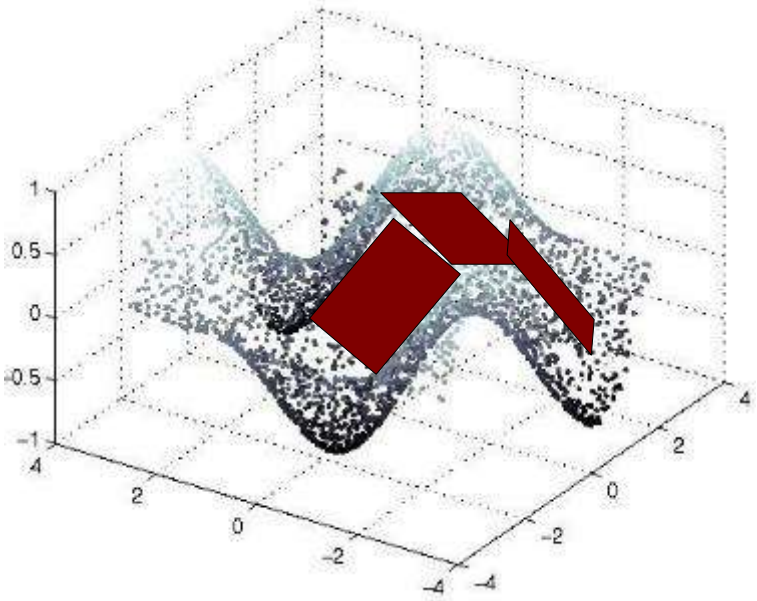
问题。为什么图层之间的映射不能是线性的？
回答：。因为线性函数的组成是一个线性函数。 神经网络将简化为（1层）逻辑回归。
问题。ReLU层的作用是什么？
解答：ReLU层实现了什么？分片线性铺设：映射是局部线性的。

Question: Why can't the mapping between layers be linear?

Answer: Because composition of linear functions is a linear function. Neural network would reduce to (1 layer) logistic regression.

Question: What do ReLU layers accomplish?

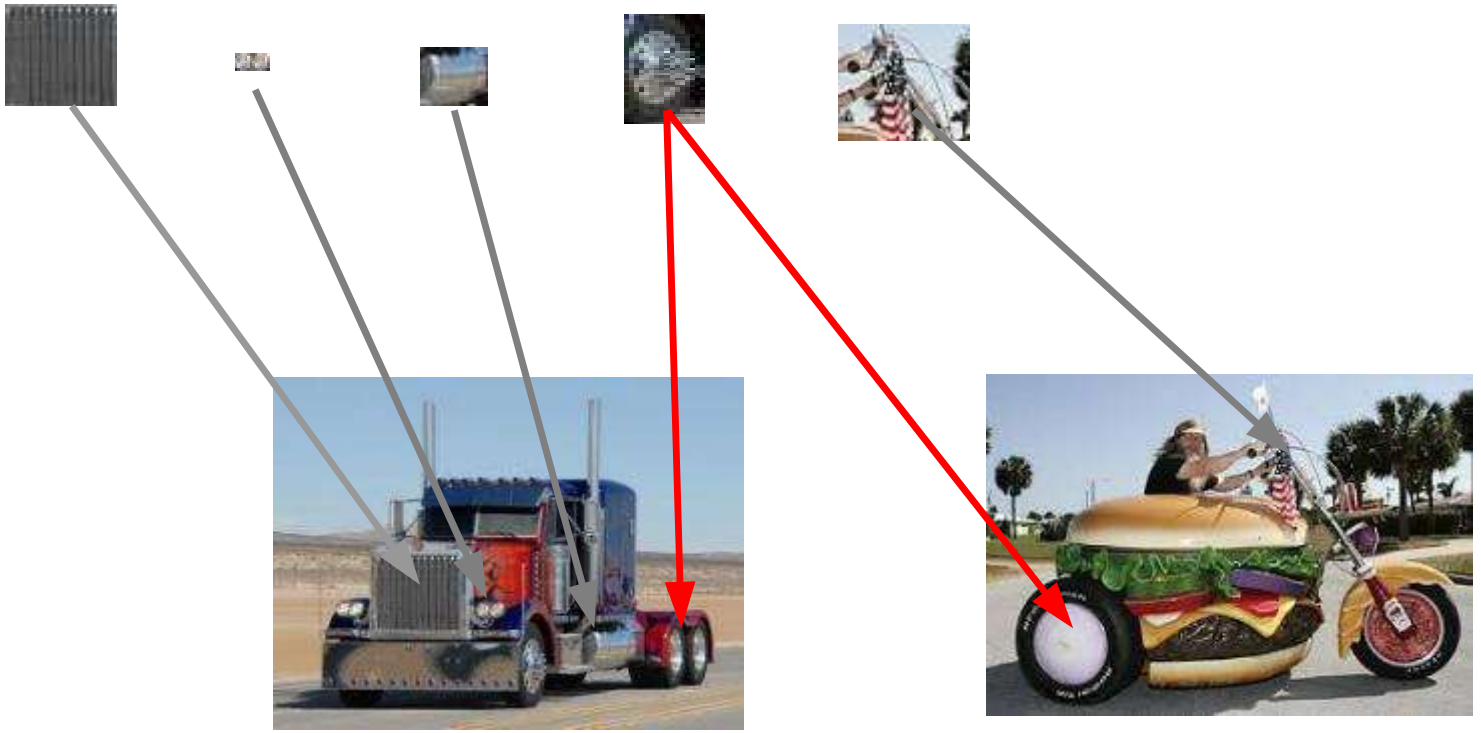
Answer: Piece-wise linear tiling: mapping is locally linear.



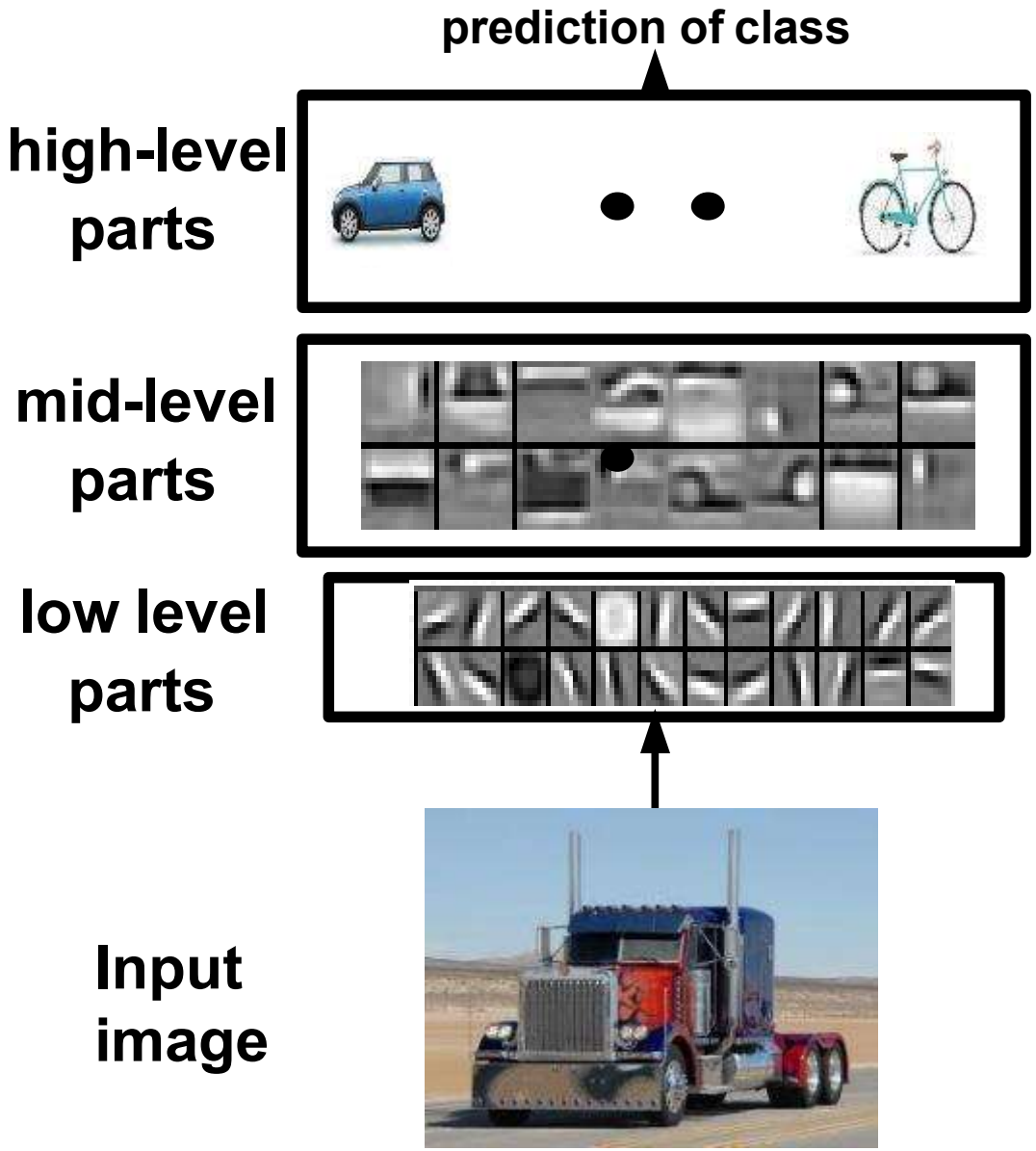
Montufar et al. "On the number of linear regions of DNNs" arXiv 2014

[1 1 0 0 0 1 0 1 0 0 0 0 1 1 0 1…] motorbike

[0 0 1 0 0 0 0 1 0 0 1 1 0 0 1 0 …] truck



Interpretation



- distributed representations feature sharing
- compositionality

Lee et al. “Convolutional DBN's ...” ICML 2009

问题。隐蔽单位是做什么的？
答: 隐蔽单元是做什么的？它可以被认为是一个分类器或特征检测器。
问题。有多少层？有多少个隐藏单元？答案。交叉验证或超参数搜索方法是答案。一般来说，网络越宽、越深，映射就越复杂。
问题。如何设置权重矩阵？
解答：权重矩阵和偏置的设置是在网络中进行的。权重矩阵和偏置是学来的。
首先，我们需要定义一个衡量当前映射质量的标准。 然后，我们需要定义一个程序来调整参数。

Question: What does a hidden unit do?

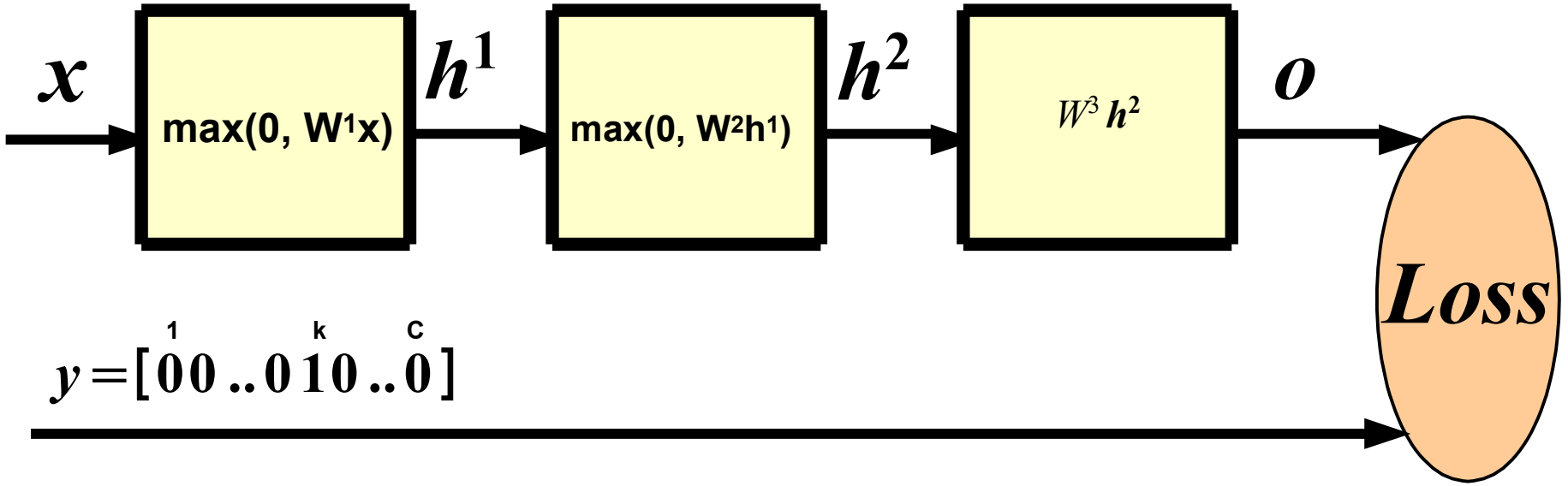
Answer: It can be thought of as a classifier or feature detector.

Question: How many layers? How many hidden units?

Answer: Cross-validation or hyper-parameter search methods are the answer. In general, the wider and the deeper the network the more complicated the mapping.

Question: How do I set the weight matrices?

Answer: Weight matrices and biases are learned.
First, we need to define a measure of quality of the current mapping. Then, we need to define a procedure to adjust the parameters.



Probability of class k given input (softmax):

$$P(c_k = 1|x) = \frac{e^{o_k}}{\sum_{j=1}^{c} e^{o_j}}$$

(Per-sample) Loss; e.g., negative log-likelihood (good for classification of small number of classes):

$$L(x, y; \theta) = -\sum_{j=1}^{c} y_j \log p(c_j|x)$$

Learning consists of minimizing the loss (plus some regularization term) w.r.t. parameters over the whole training set.

$$\theta^* = \text{argmin}_\theta \sum_{n=1}^{N} L(x^n, y^n; \theta)$$

Question: How to minimize a complicated function of the parameters?

Answer: Chain rule, a.k.a. **Backpropagation**! That is the procedure to compute gradients of the loss w.r.t. parameters in a multi-layer neural network.

Rumelhart et al. "Learning internal representations by back-propagating.." Nature 1986

$$P(c_k = 1|x) = \frac{e^{o_k}}{\sum_{j=1}^{c} e^{o_j}}$$

$$L(x, y; \theta) = -\sum_{j=1}^{c} y_j \log p(c_j|x)$$

$$y = [\overset{1}{0}0 .. 0\overset{k}{1}0 .. \overset{C}{0}]$$

By substituting the first formula in the second, and taking the derivative w.r.t. O we get:

$$\frac{\partial L}{\partial o} = p(c|x) - y$$

5 mins: prove it!

$$P(c_k = 1|x) = \frac{e^{o_k}}{\sum_{j=1}^{c} e^{o_j}}$$

$$L(x, y; \theta) = -\sum_{j=1}^{c} y_j \log p(c_j|x)$$

$$y = [\overset{1}{0}0 \,..\, 0\overset{k}{1}0 \,..\, 0\overset{C}{}]$$

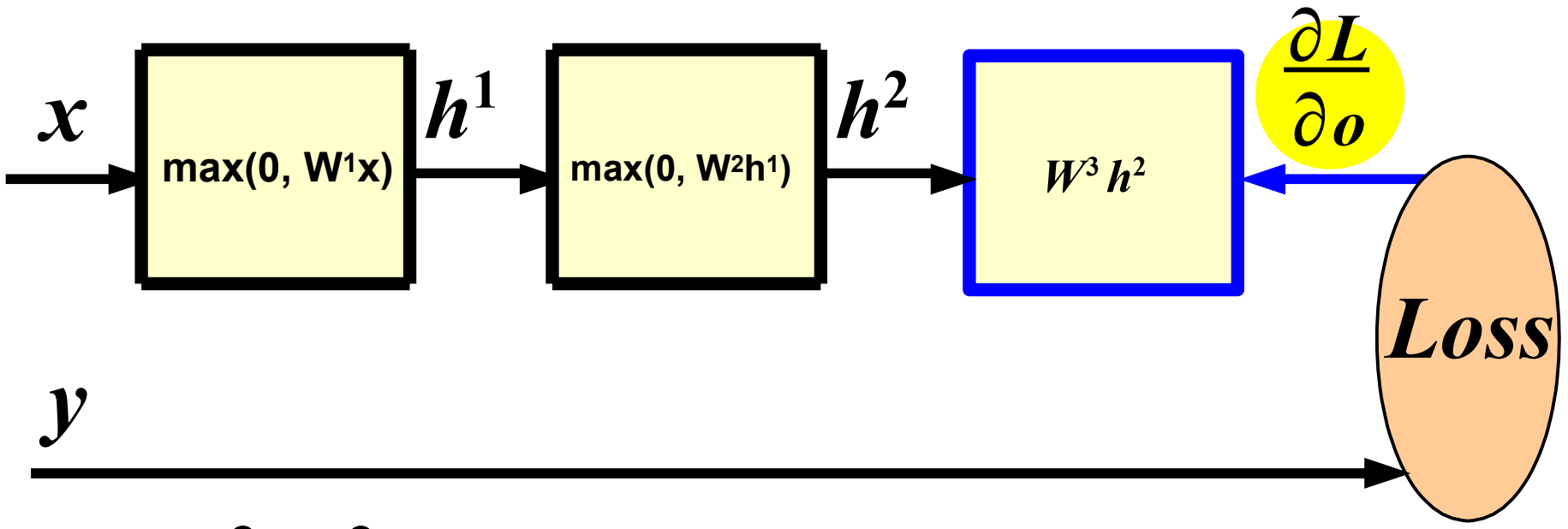
Since only y_k = 1 and other y_i's are equal to 0.

$$L(x, y; \theta) = -\log p(c_k = 1|x) = -\log \frac{e^{o_k}}{\sum_{j=1}^{c} e^{o_j}} = -o_k + \log \sum_{j=1}^{c} e^{o_j}$$

$$\frac{\partial L}{\partial o_k} = -1 + \frac{\partial}{\partial o_k} \log \sum_{j=1}^{c} e^{o_j} = -1 + \frac{\partial \sum_{j=1}^{c} e^{o_j} / \partial o_k}{\sum_{j=1}^{c} e^{o_j}} = -1 + \frac{e^{o_k}}{\sum_{j=1}^{c} e^{o_j}}$$

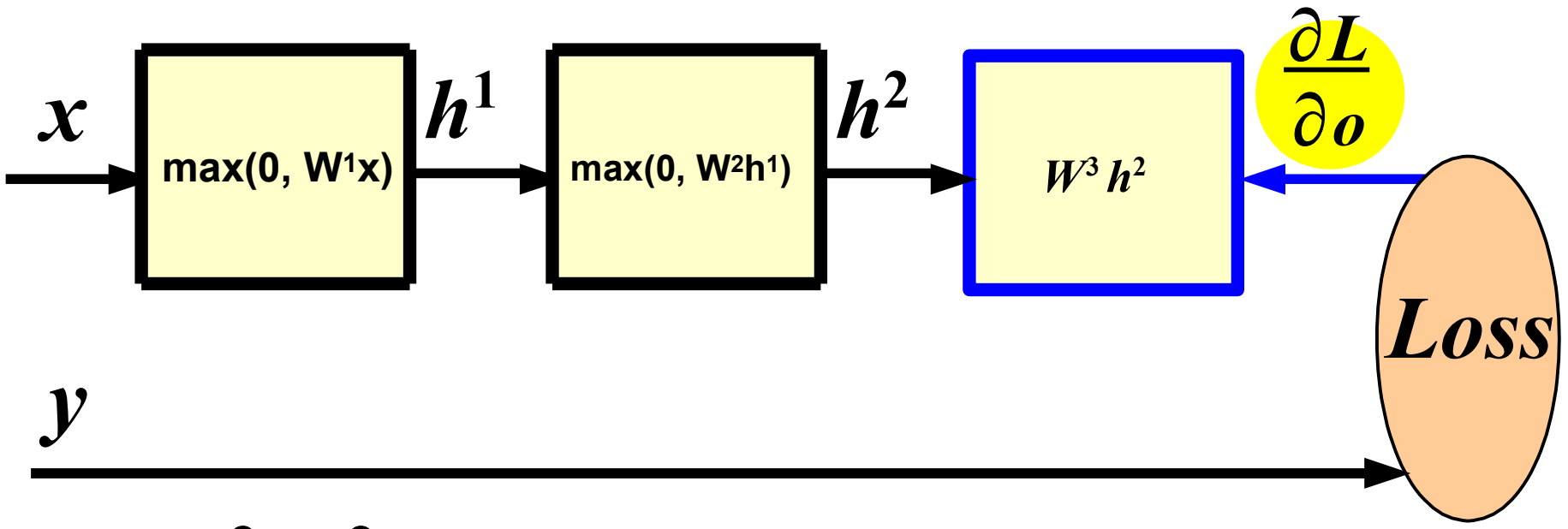
$$\frac{\partial L}{\partial o_j} = 0 + \frac{\partial}{\partial o_j} \log \sum_{j=1}^{c} e^{o_j} = -1 + \frac{\partial \sum_{j=1}^{c} e^{o_j} / \partial o_j}{\sum_{j=1}^{c} e^{o_j}} = 0 + \frac{e^{o_k}}{\sum_{j=1}^{c} e^{o_j}} \quad \text{For } j \neq k$$

$$\frac{\partial L}{\partial o} = p(c|x) - y$$



Given $\partial L / \partial o$ and assuming we can easily compute the Jacobian of each module, we have:

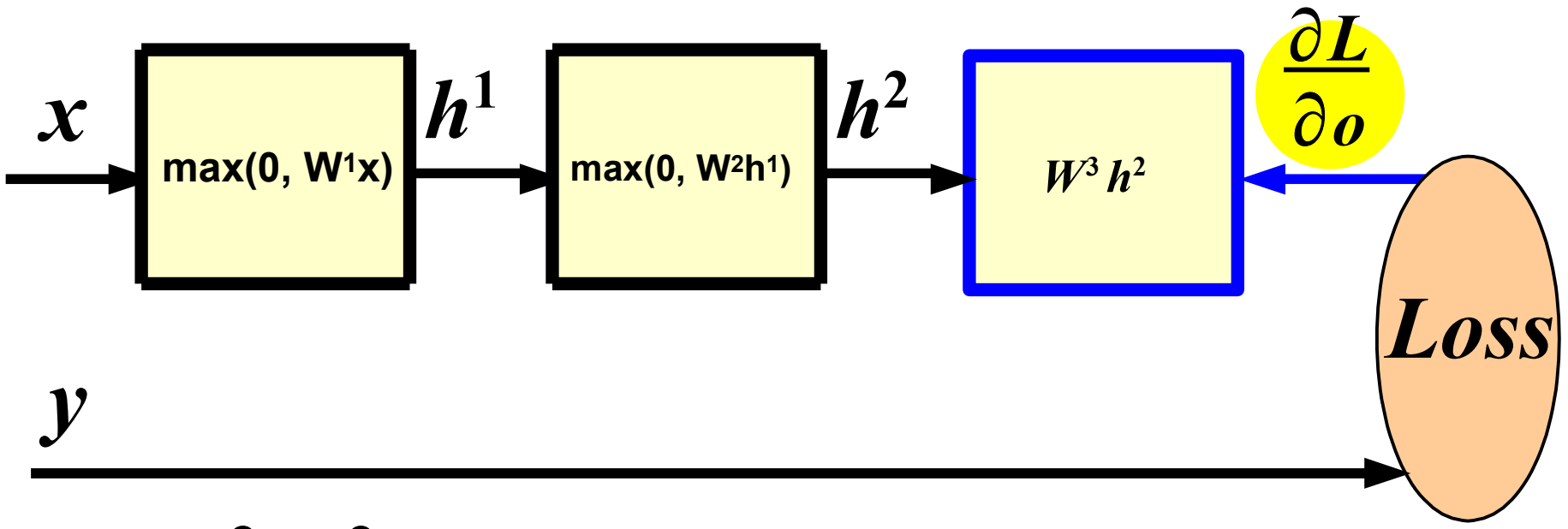
$$\frac{\partial L}{\partial W^3} = \frac{\partial L}{\partial o} \frac{\partial o}{\partial W^3}$$



Given $\partial L/\partial o$ and assuming we can easily compute the Jacobian of each module, we have:

$$\frac{\partial L}{\partial W^3} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial W^3}$$

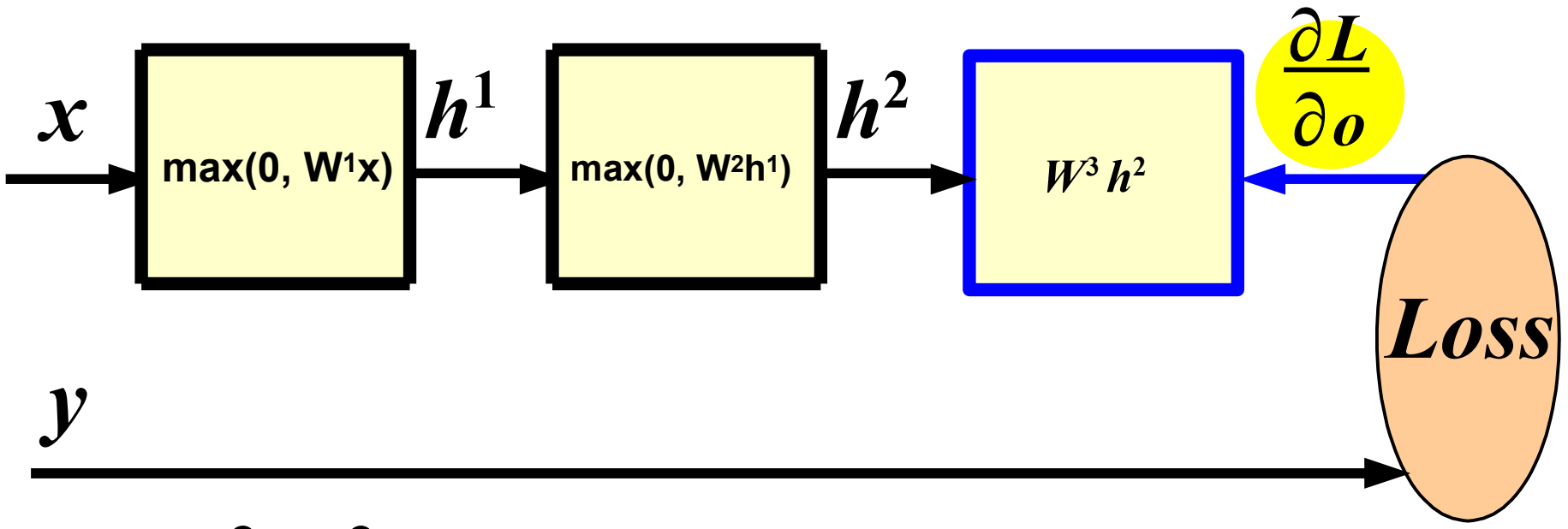
$$\frac{\partial L}{\partial W^3} = (p(c|x) - y)h^{2T}$$



Given $\partial L/\partial o$ and assuming we can easily compute the Jacobian of each module, we have:

$$\frac{\partial L}{\partial W^3} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial W^3} \qquad \frac{\partial L}{\partial h^2} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial h^2}$$

$$\frac{\partial L}{\partial W^3} = (p(c|x) - y)h^{2T}$$

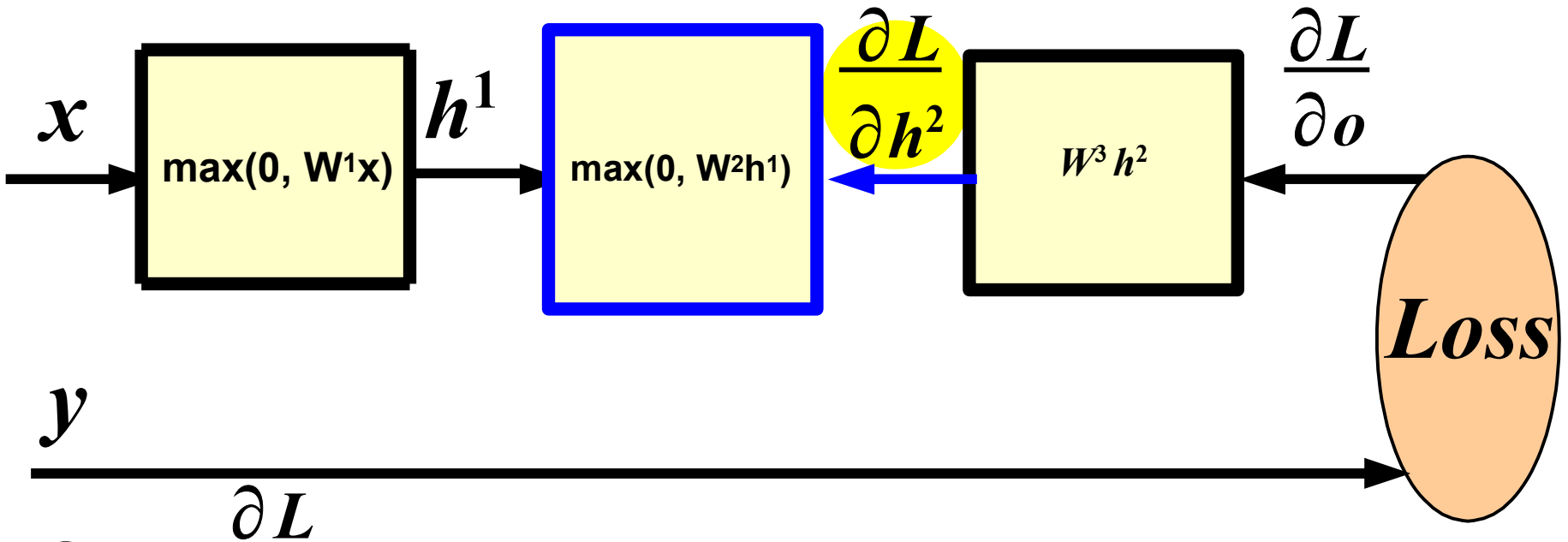


Given $\partial L/\partial o$ and assuming we can easily compute the Jacobian of each module, we have:

$$\frac{\partial L}{\partial W^3} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial W^3} \qquad \frac{\partial L}{\partial h^2} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial h^2}$$

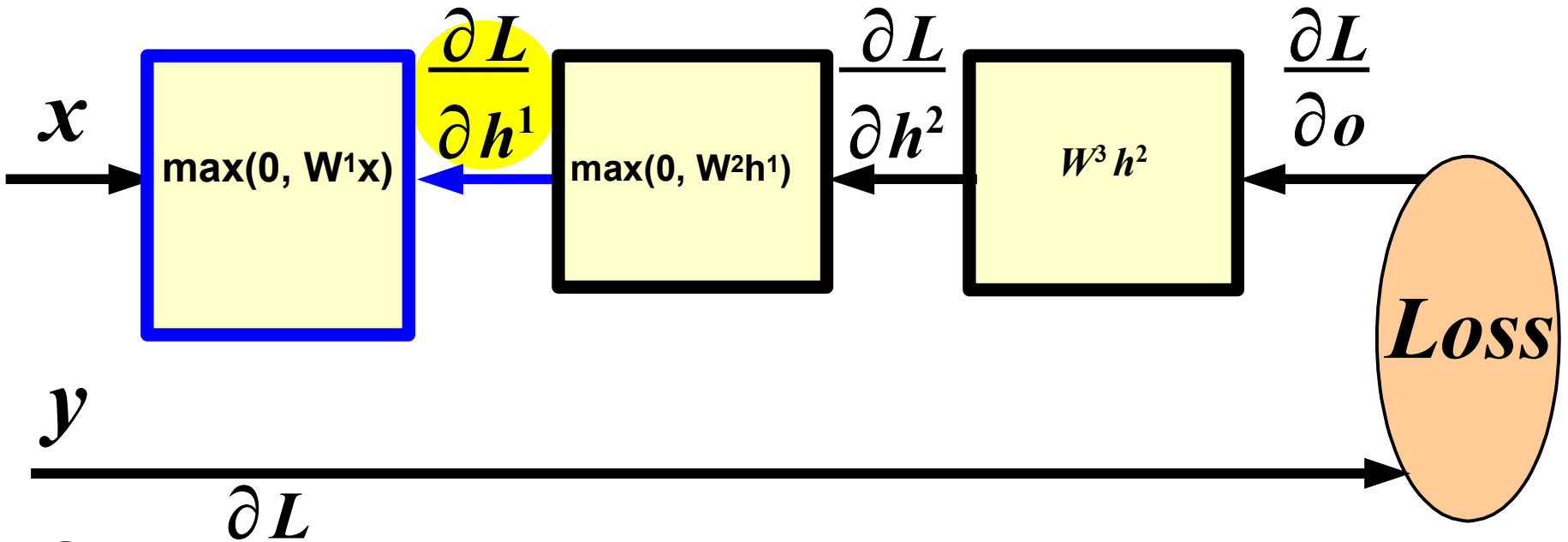
$$\frac{\partial L}{\partial W^3} = (\boldsymbol{p}(\boldsymbol{c}|\boldsymbol{x}) - \boldsymbol{y})\boldsymbol{h}^{\boldsymbol{2T}} \qquad \frac{\partial L}{\partial h^2} = \boldsymbol{W}^{\boldsymbol{3T}}(\boldsymbol{p}(\boldsymbol{c}|\boldsymbol{x}) - \boldsymbol{y})$$

Backward Propagation



Given $\frac{\partial L}{\partial h^2}$ we can compute now:

$$\frac{\partial L}{\partial W^2} = \frac{\partial L}{\partial h^2}\frac{\partial h^2}{\partial W^2} \qquad \frac{\partial L}{\partial h^1} = \frac{\partial L}{\partial h^2}\frac{\partial h^2}{\partial h^1}$$



Given $\frac{\partial L}{\partial h^1}$ we can compute now:

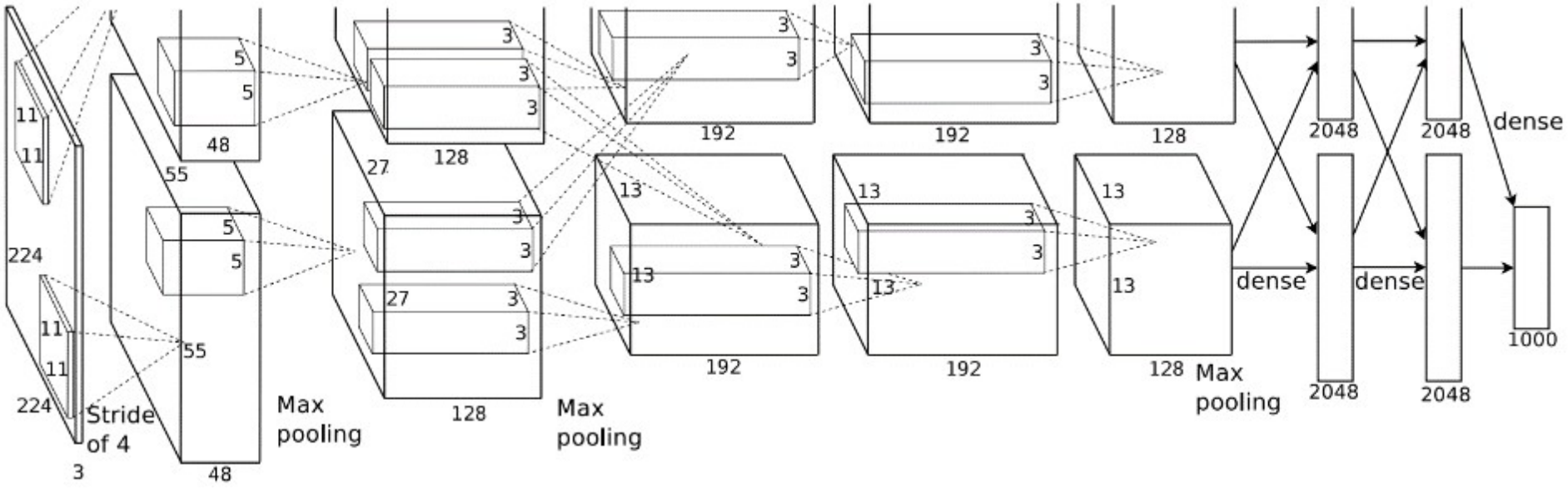
$$\frac{\partial L}{\partial W^1} = \frac{\partial L}{\partial h^1}\frac{\partial h^1}{\partial W^1}$$

Applications

- Audio & Speech
  - Speaker identification, Phone classification, etc.
- Language & NLP
  - Language modeling, tagging, named entity recognition, semantic role labeling, etc.
- Vision
  - Object classification, image retrieval, action recognition, face parsing, etc.
- Others

Vision

- Deep CNN on ImageNet Classification (Krizhevsky et al. 2012)



Overall architecture of the Deep CNN Model

Vision

- Deep CNN on ImageNet Classification (Krizhevsky et al. 2012)
  - 7 hidden layers not counting max pooling.
  - Early layers are convolutional, last two layers are globally connected.
  - Use “dropout” technique to regularize the weights in the globally connected layers.
  - Uses data augmentation method to reduce over fitting。
  - Trained on random 224x224 patches from 256x256 images to get more data.
  - Takes between five and six days to train on two GTX 580 3GB GPUs.

Vision

- Deep CNN on ImageNet Classification (Krizhevsky et al. 2012)

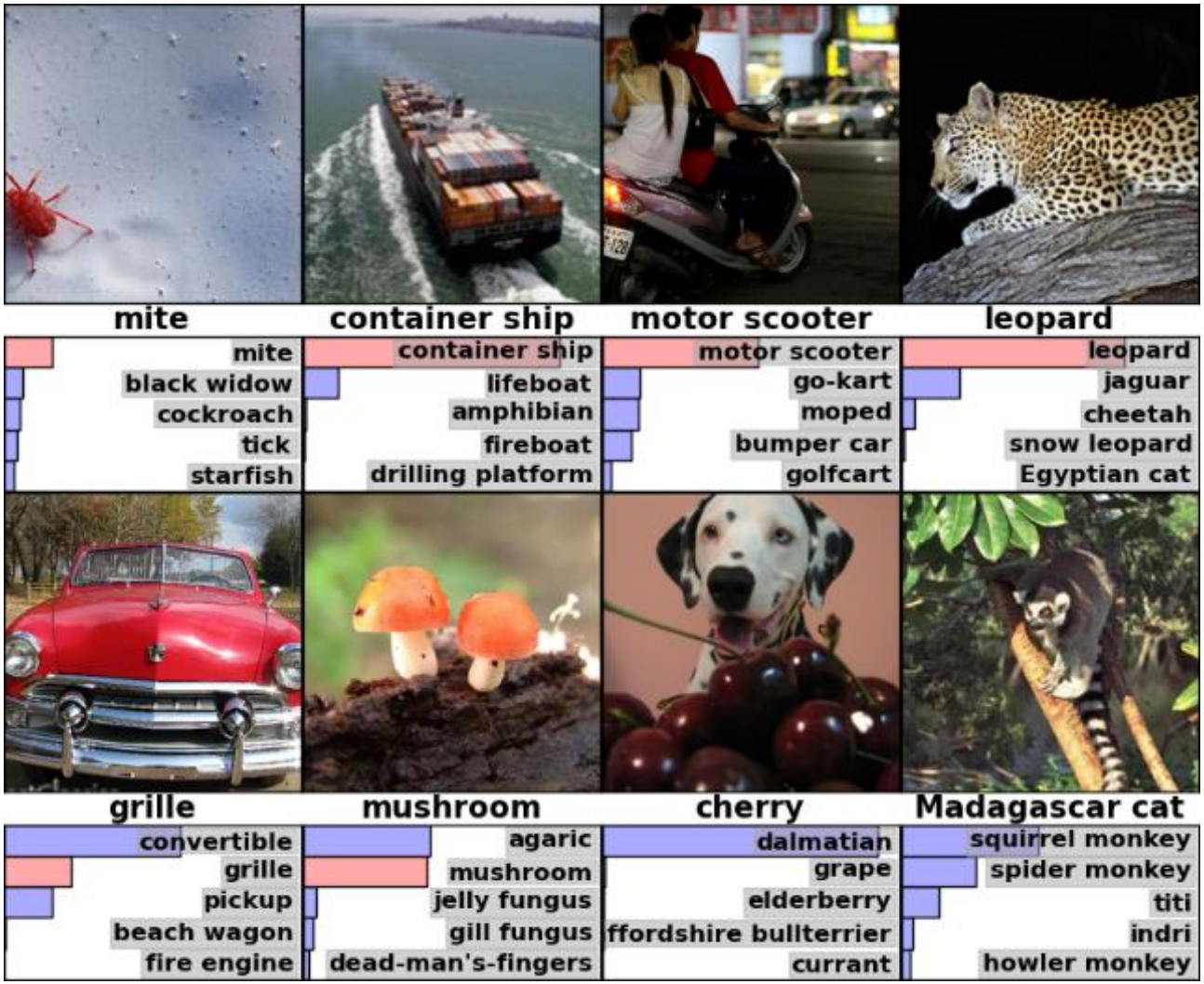
72%, 2010

74%, 2011

85%, 2012

Vision

- Deep CNN on ImageNet Classification (Krizhevsky et al. 2012)



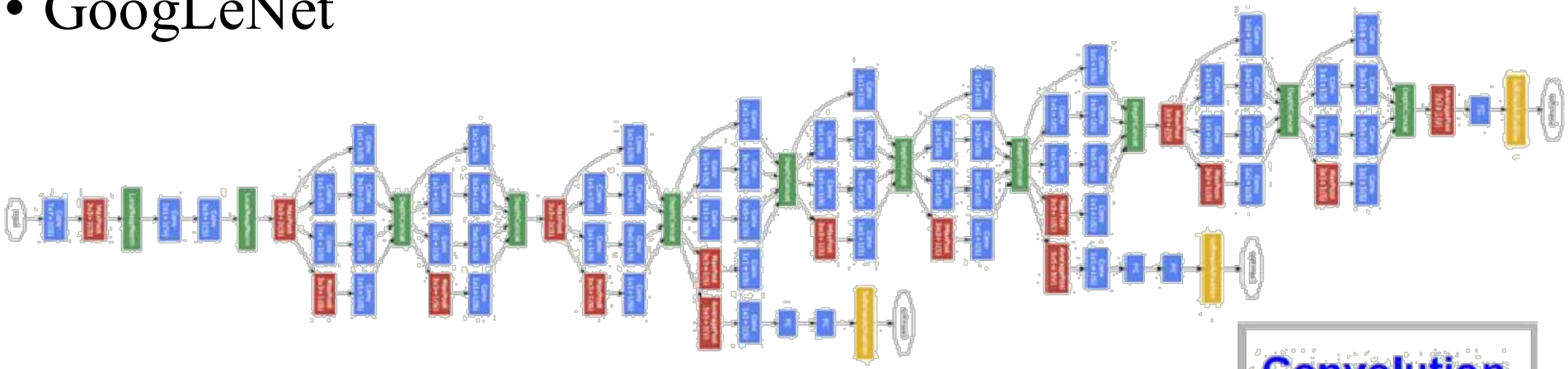
Vision

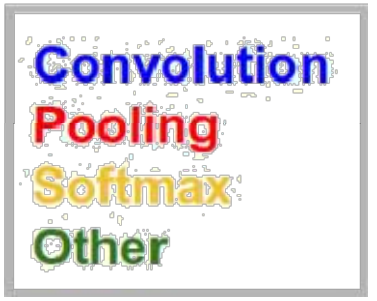
- Deep CNN on ImageNet Classification (Krizhevsky12)

What's next?

Vision

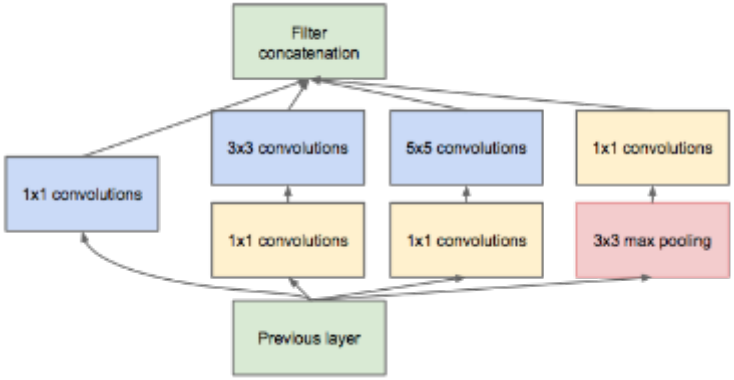
- GoogLeNet



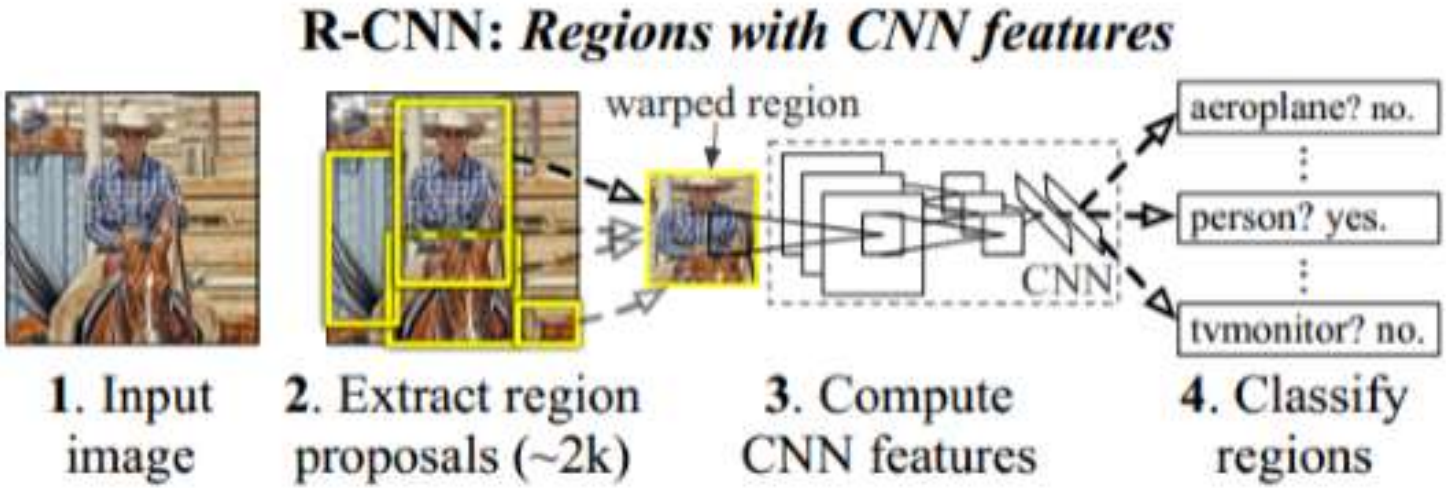


ILSVRC14 Winners: ~6.6% Top-5 error

- **GoogLeNet: composition of multi-scale dimension-reduced modules (pictured)**
- **VGG: 16 layers of 3x3 convolution interleaved with max pooling + 3 fully-connected layers**



R-CNN for Object Detection



- Use Selective Search to get the region proposals
- For each proposal, use CNN to get features (from the fc7 of AlexNet)
- Input the feature to learn the classifiers to predict scores
- Rank the scores to get the final solution

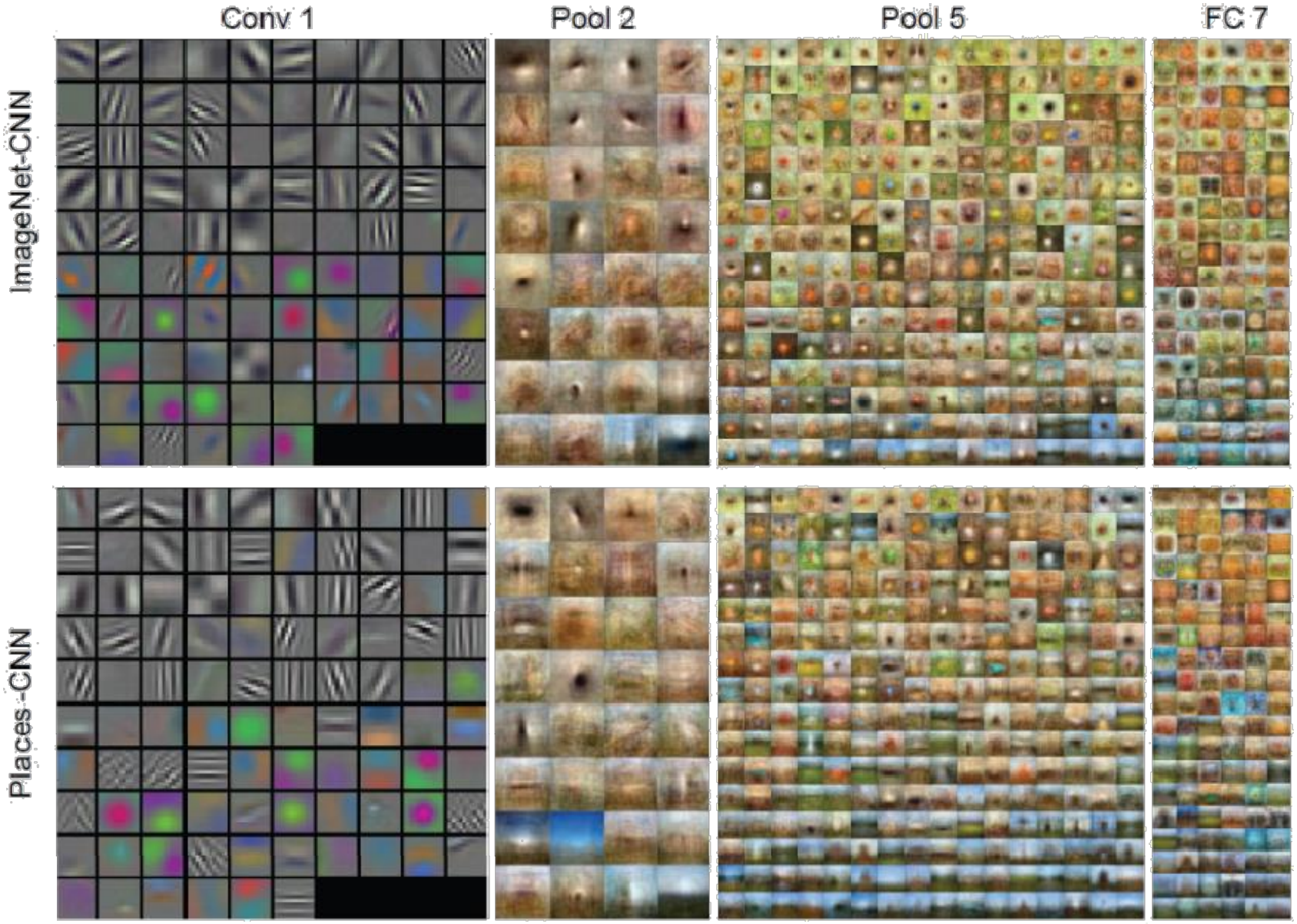
- 使用选择性搜索来获得区域建议
- 对于每个提议，使用CNN来获取特征（来自AlexNet的fc7）。
- 输入特征来学习分类器来预测分数
- 对分数进行排序，得到最终的解决方案

Learning Deep Features for Scene Recognition using Places Database

- Zhou et al., NIPS 2014

Image samples from the scene categories grouped by their queried adjectives.

Learning Deep Features for Scene Recognition using Places Database



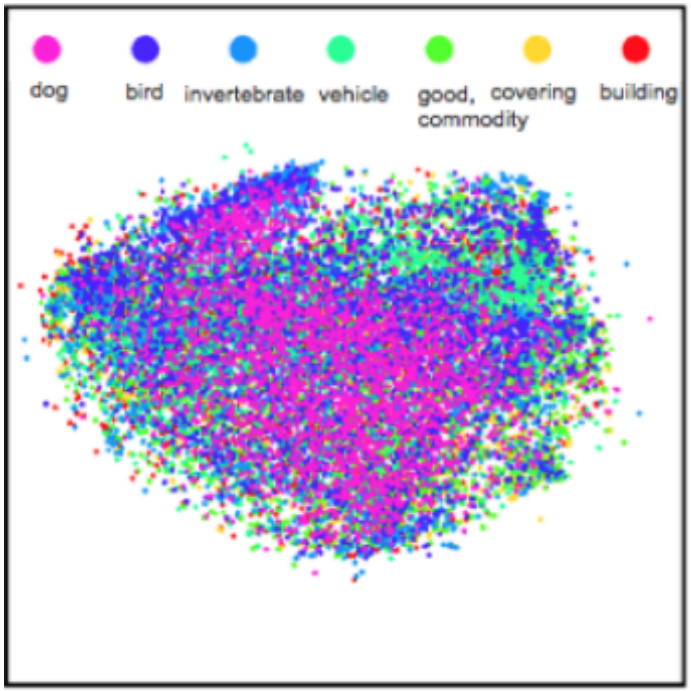
Learning Deep Features for Scene Recognition using Places Database

Classification accuracy/precision on scene-centric databases and object-centric databases for the Places-CNN feature and ImageNet-CNN feature. The classifier in all the experiments is a linear SVM with the same parameters for the two features.

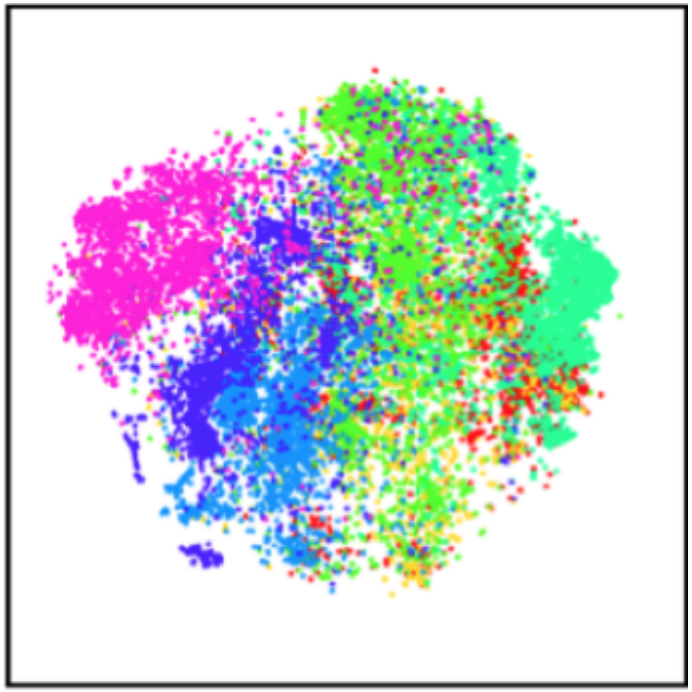
	SUN397	MIT Indoor67	Scene15	SUN Attribute
Places-CNN feature	**54.32±0.14**	**68.24**	**90.19±0.34**	**91.29**
ImageNet-CNN feature	42.61±0.16	56.79	84.23±0.37	89.85
	Caltech101	Caltech256	Action40	Event8
Places-CNN feature	65.18±0.88	45.59±0.31	42.86±0.25	94.12±0.99
ImageNet-CNN feature	**87.22±0.92**	**67.23±0.27**	**54.92±0.33**	**94.42±0.76**

Why Deep Learning?

- The Unreasonable Effectiveness of Deep Features



Low-level: $Pool_1$

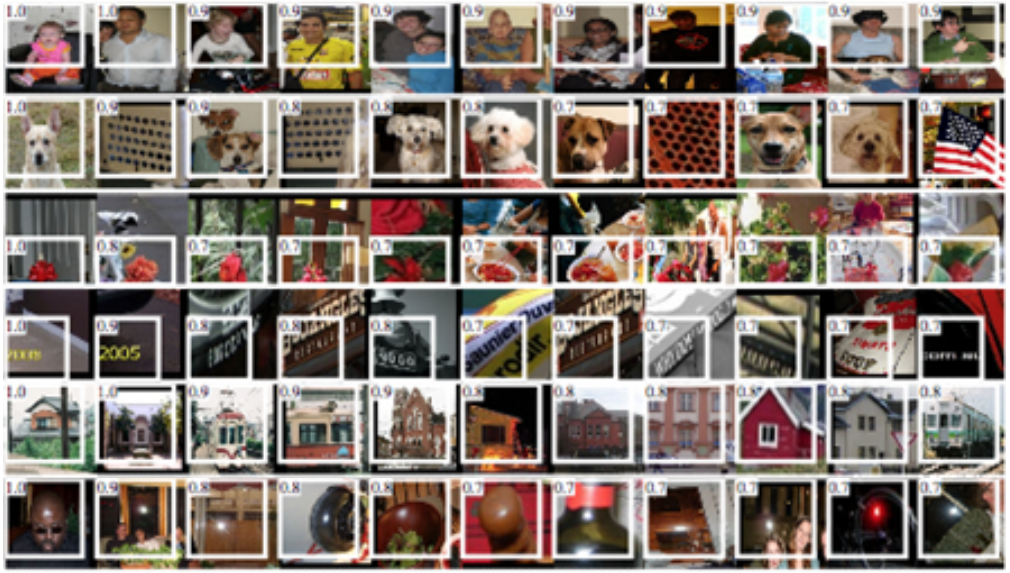


High-level: FC_6

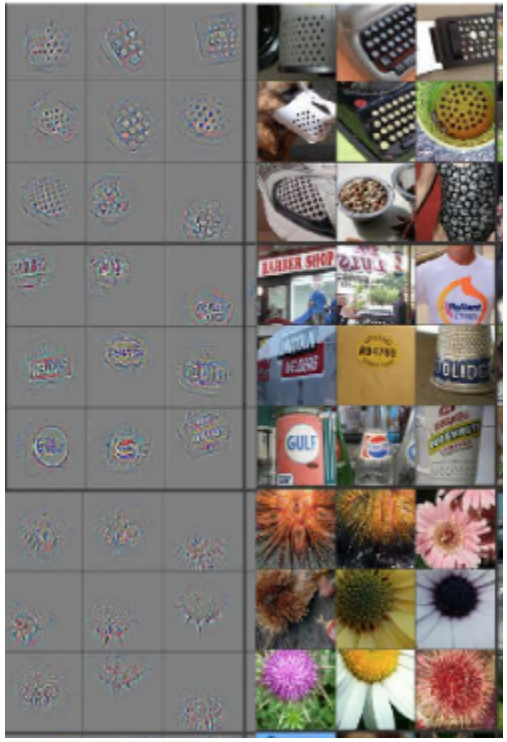
Classes separate in the deep representations and transfer to many tasks.
[DeCAF] [Zeiler-Fergus]

Why Deep Learning?

- Rich visual structure of features deep in hierarchy.



Maximal activations of $pool_5$ units **[R-CNN]**



$conv_5$ DeConv visualization
[Zeiler-Fergus]

Why Deep Learning?

- Visualization

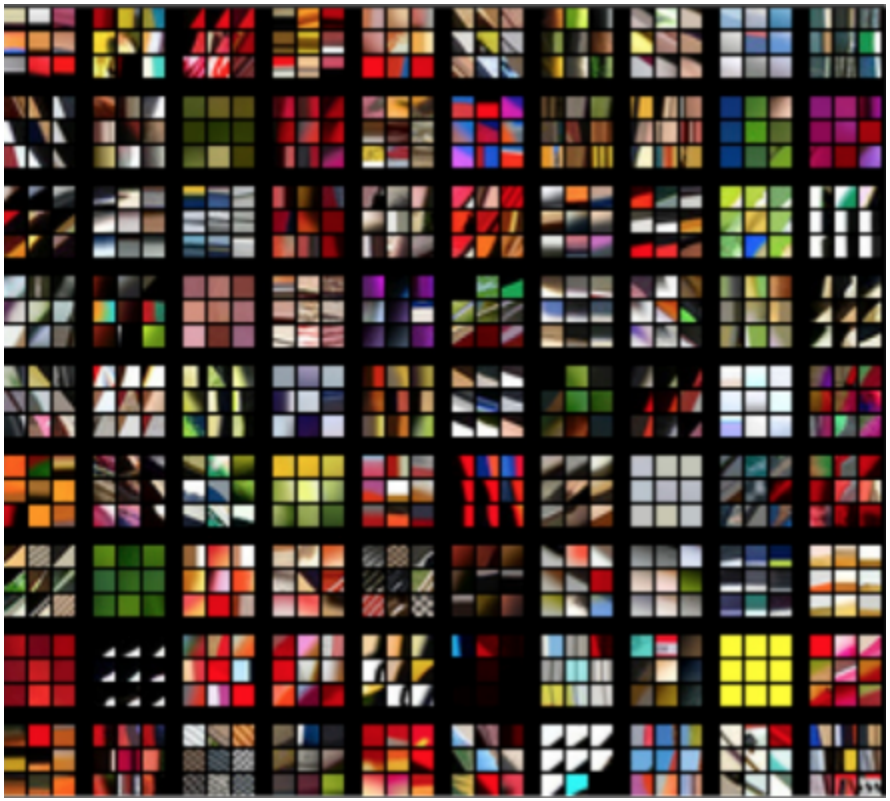
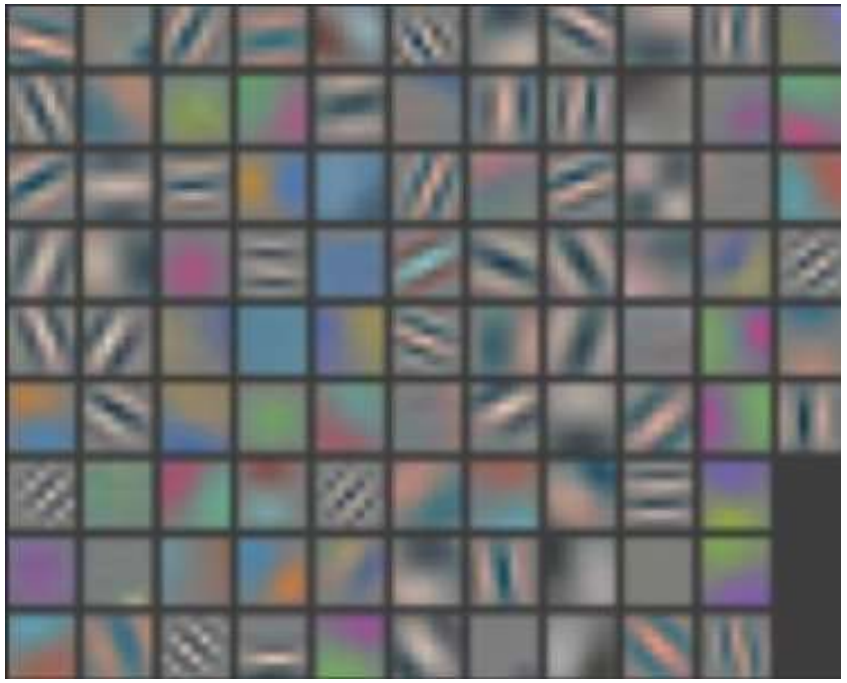


image patches that strongly activate 1st layer filters



1st layer filters

[Zeiler-Fergus]

Choosing The Architecture

- **Task dependent**
- **Cross-validation**
- **[Convolution → Normalization → pooling]* + fc layer**
- **The more data: the more layers and the more kernels**
  - **Look at the number of parameters at each layer**
  - **Look at the number of flops at each layer**
- **Computational resources**
- **Be creative :)**

How To Optimize

- **SGD (with momentum) usually works very well**
- **Pick learning rate by running on a subset of the data**
  Bottou "Stochastic Gradient Tricks" Neural Networks 2012
  - **Start with large learning rate and divide by 2 until loss does not diverge**
  - **Decay learning rate by a factor of ~1000 or more by the end of training**
- **Use  non-linearity**
- **Initialize parameters so that each feature across layers has similar variance. Avoid units in saturation.**

Improving Generalization

- Weight sharing (greatly reduce the number of parameters)
- Data augmentation (e.g., jittering, noise injection, etc.)
- Dropout

  Hinton et al. “Improving Nns by preventing co-adaptation of feature detectors” arxiv 2012
- Weight decay (L2, L1)
- Sparsity in the hidden units
-

ConvNets: today

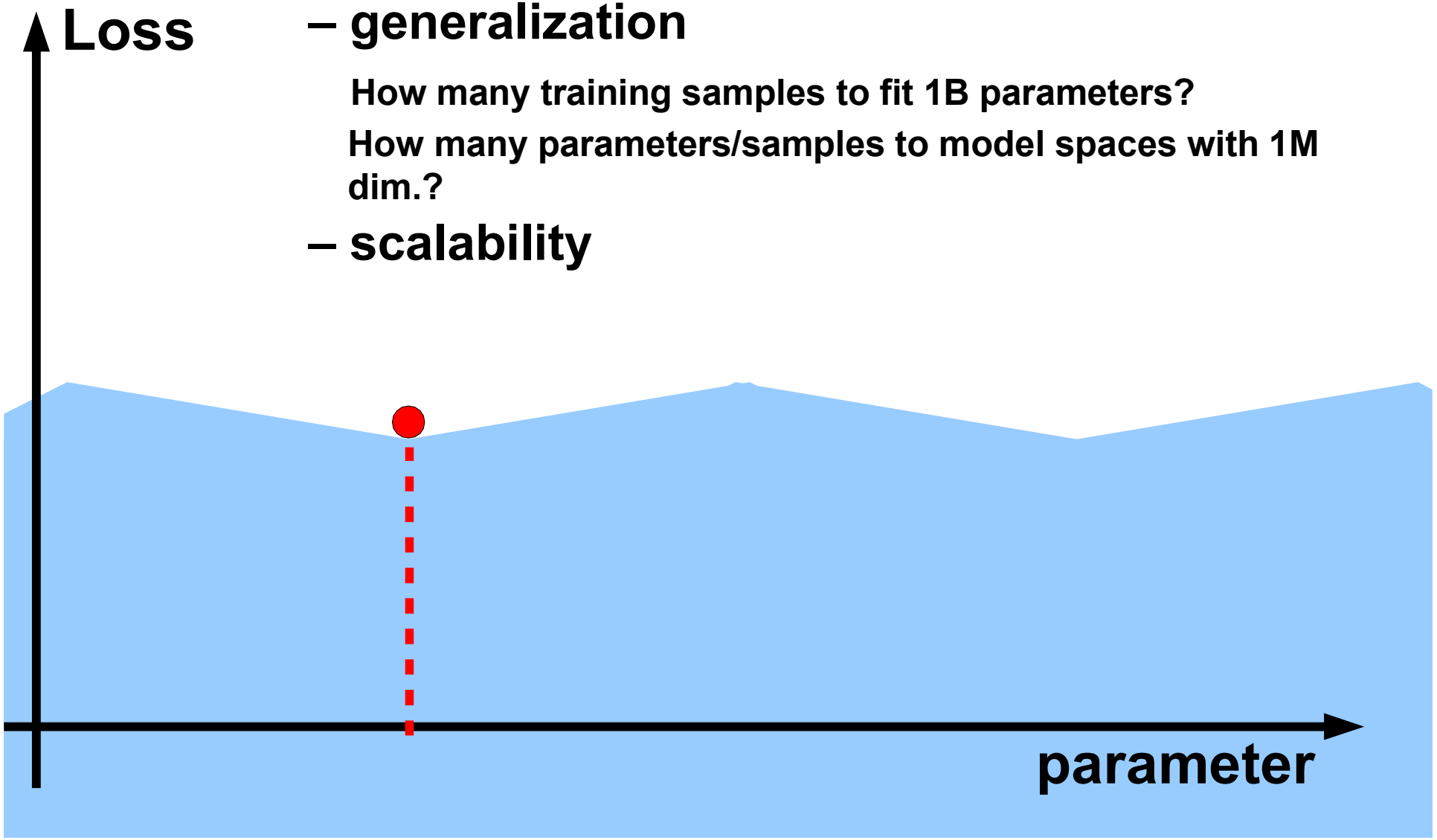
Today's belief is that the challenge is about:

– generalization

How many training samples to fit 1B parameters?

How many parameters/samples to model spaces with 1M dim.?

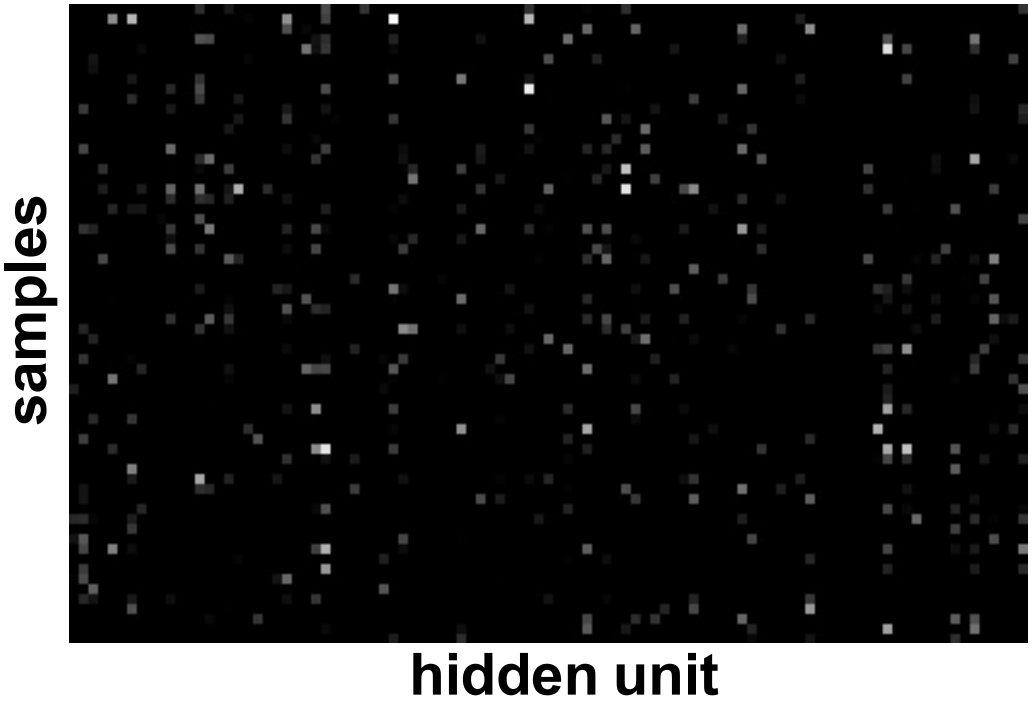
– scalability



Good To Know

值得了解的是
通过有限差分数值检查梯度
可视化特征（特征图需要不
相关），并具有高方差。

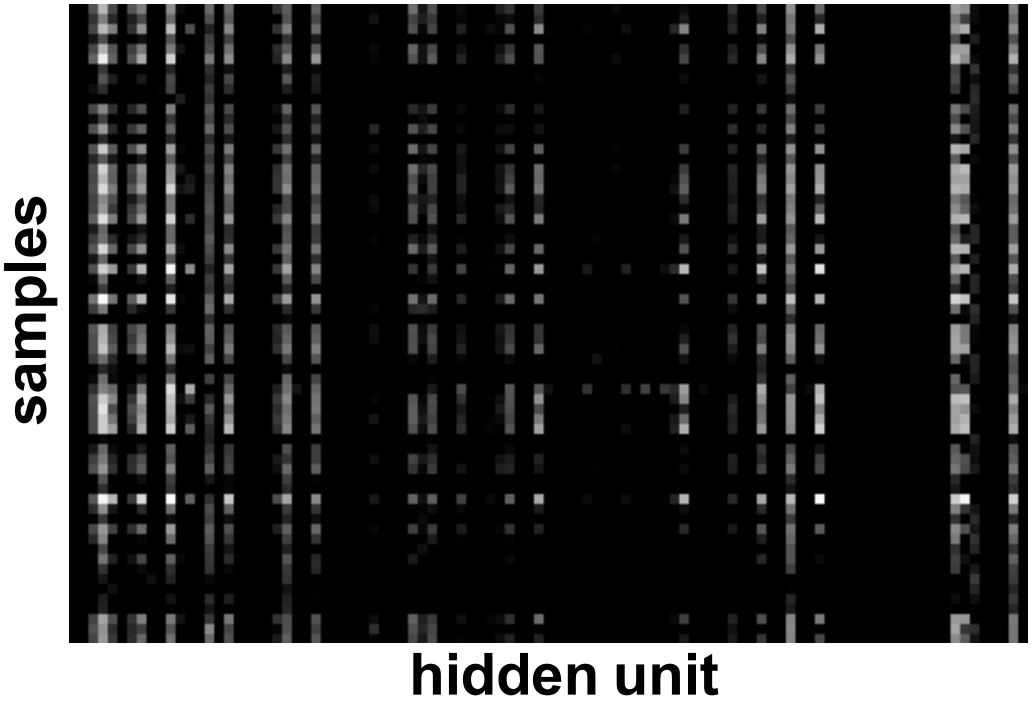
- **Check gradients numerically by finite differences**
- **Visualize features (feature maps need to be uncorrelated and have high variance.**



Good training: hidden units are sparse across samples and across features.

Good To Know

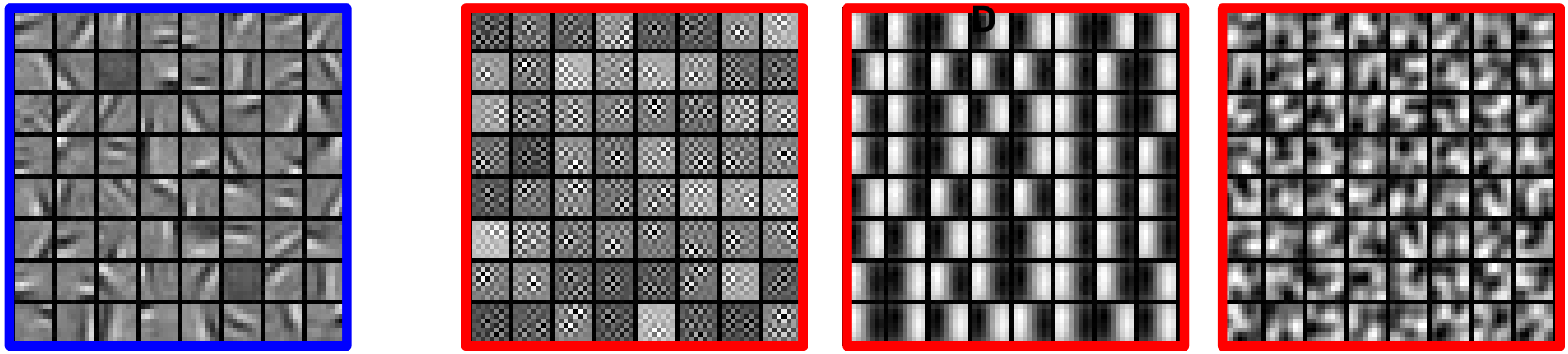
- **Check gradients numerically by finite differences**
- **Visualize features (feature maps need to be uncorrelated and have high variance.**



Bad training: many hidden units ignore the input and/or exhibit strong correlations.

Good To Know

- Check gradients numerically by finite differences
- Visualize features (feature maps need to be uncorrelated and have high variance.
- Visualize parameters



Good training: learned filters exhibit structure and are uncorrelated.

Zeiler, Fergus "Visualizing and understanding CNNs" arXiv 2013
Simonyan, Vedaldi, Zisserman "Deep inside CNNs: visualizing image classification models.." ICLR 2014

Good To Know

- **Check gradients numerically by finite differences**
- **Visualize features (feature maps need to be uncorrelated and have high variance.**
- **Visualize parameters**
- **Measure error on both training and validation set.**
- **Test on a small subset of the data and check the error → 0.**

What If It Does Not

- **Training diverges:**
  - **Learning rate may be too large → decrease learning rate**
  - **BPROP is buggy → numerical gradient checking**
- **Parameters collapse / loss is minimized but accuracy is low**
  - **Check loss function:**
    - **Is it appropriate for the task you want to solve?**
    - **Does it have degenerate solutions? Check "pull-up" term.**
- **Network is underperforming**
  - **Compute flops and nr. params. →if too small, make net larger**
  - **Visualize hidden units/params → fix optmization**
- **Network is too slow**
  - **Compute flops and nr. params.**
    → GPU,distrib. framework, make net smaller

Summary

- Key characteristics of existing Deep Learning models
  - Deep architectures=feature hierarchy (>= 2 hidden layers)
  - Unsupervised pre-training from a large number of samples to build a good initialization of the model
  - Supervised fine-tuning to further improve the performance
  - Significant performance improvements over previous methods on many applications
- Challenges of Deep Learning
  - It is still lacking of some solid theoretical guarantees
  - It has many tricks to play with

- 现有深度学习模型的主要特征
- 深度架构=特征层次(>=2个隐藏层)
- 从大量的样本中进行无监督的预训练，以建立良好的模型初始化
- 监督下的微调以进一步提高性能
- 在许多应用上比以前的方法有明显的性能改进
- 深度学习的挑战
- 它仍然缺乏一些坚实的理论保证
- 它有很多技巧可供玩味

Deep Learning Reading List

- Survey and Tutorials
  - Representation Learning: A Review and New Perspectives, Yoshua Bengio, Aaron Courville, Pascal Vincent, Arxiv, 2012.
  - The monograph or review paper Learning Deep Architectures for AI (Foundations & Trends in Machine Learning, 2009).
  - Deep Machine Learning – A New Frontier in Artificial Intelligence Research – a survey paper by Itamar Arel, Derek C. Rose, and Thomas P. Karnowski.
- Papers
  - G. E. Hinton, and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science,* vol. 313, no. 5786, pp. 504-7, Jul 28, 2006.
  - ……
- Codes and Demos
  - Theano, Pylearn2, DeepLearnToolbox, etc.
  - http://caffe.berkeleyvision.org/
- Internet Resourses
  - http://deeplearning.net/
  - https://sites.google.com/site/representationlearning2014/

References

1. Honglak Lee, Yan Largman, Peter Pham, and Andrew Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In NIPS, 2009.
2. George E. Dahl, Marc'Aurelio Ranzato, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Phone Recognition with the Mean-Covariance Restricted Boltzmann Machine. In NIPS 2010.
3. Abdel-rahman Mohamed, Tara N. Sainath, George E. Dahl, Bhuvana Ramabhadran, Geoffrey E. Hinton, and Michael A. Picheny, Deep Belief Networks Using Discriminative Features for Phone Recognition, In ICASSP, 2011.
4. R. Collobert and J. Weston. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In ICML, 2008.
5. Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. and A. Ng. Building High-Level Features using Large Scale Unsupervised Learning. In ICML 2012.
6. A. Krizhevsky, I. Sutskever. and G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In NIPS 2012.
7. A. Krizhevsky and G. Hinton. Using Very Deep Autoencoders for Content-Based Image Retrieval. In ESANN, 2011.
8. Q. Le, W. Zou, S. Yeung and A. Ng. Learning Hierarchical Spatio-temporal Features for Action Recognition with Independent Subspace Analysis. In CVPR 2011.